A DISCRETE ADAPTED HIERARCHICAL BASIS SOLVER FOR RADIAL BASIS FUNCTION INTERPOLATION

JULIO E. CASTRILLÓN-CANDÁS *, JUN LI †, AND VICTOR EIJKHOUT ‡

Abstract. In this paper we develop a discrete Hierarchical Basis (HB) to efficiently solve the Radial Basis Function (RBF) interpolation problem with variable polynomial order. The HB forms an orthogonal set and is adapted to the kernel seed function and the placement of the interpolation nodes. Moreover, this basis is orthogonal to a set of polynomials up to a given order defined on the interpolating nodes. We are thus able to decouple the RBF interpolation problem for any order of the polynomial interpolation and solve it in two steps: (1) The polynomial orthogonal RBF interpolation problem is efficiently solved in the transformed HB basis with a GMRES iteration and a diagonal, or block SSOR preconditioner. (2) The residual is then projected onto an orthonormal polynomial basis. We apply our approach on several test cases to study its effectiveness, including an application to the Best Linear Unbiased Estimator regression problem.

Key words. Radial Basis Function, Interpolation, Hierarchical Basis, Integral equations, Fast Summation Methods, Stable Completion, Lifting, Generalized Least Squares, Best Linear Unbiased Estimator

AMS subject classifications. 65D05,65D07, 65F25, 65F10, 62J05, 41A15

1. Introduction.

The computational cost for extracting RBF representations can be prohibitively expensive for even a moderate amount of interpolation nodes. For an N-point interpolation problem using direct methods it requires $O(N^2)$ memory and $O(N^3)$ computational cost. Moreover, since many of the most accurate RBFs have globally supported and increasing kernels, this problem is often badly conditioned and difficult to solve with iterative methods. In this paper we develop a fast, stable and memory efficient algorithm to solve the RBF interpolation problem based on the construction of a discrete HB.

Development of RBF interpolation algorithms has been widely studied in scientific computing. In general, current fast solvers are not yet optimal. One crucial observation of the RBF interpolation problem is that it can be posed as a discrete form of an integral equation. This insight allows us to extend the techniques originally introduced for integral equations to the efficient solution of RBF interpolation problems.

RBF interpolation has been studied for several decades. In 1977 Duchon [23] introduced one of the most well known RBFs, the thin-plate spline. This RBF is popular in practice since it leads to minimal energy interpolant between the interpolation nodes in 2D. In [24] Franke studied the approximation capabilities of a large class of RBFs and concluded that the biharmonic spline and the multiquadric give the best approximation. Furthermore, error estimates for RBF interpolation have been developed by Schaback et al. [56, 50, 51] and more recently by Narcowich et al. [38].

RBFs are of much interest in the area of visualization and animation. They have found applications to point cloud reconstructions, denoising and repairing of meshes [14]. In general, they have been used for the reconstruction of 3-D objects

* King Abdullah University of Science and Technology, Kingdom of Saudi Arabia, julio.castrillon@kaust.edu.sa. This author was supported in part by the Institute for Computational Engineering and Sciences Postdoctoral Fellowship at the University of Texas at Austin

† JLi49@slb.com This author was supported in part by grant # NIH R01 GM074258.

‡ eijkhout@tacc.utexas.edu

and deformation of these objects [60]. Although we notice that for these areas of applications it is usually sufficient to consider zero- and linear-order polynomials in the RBF problems. Popular applications, such as Neural Networks and classification [46], boundary and finite element methods [21, 22], require consideration of higher-order polynomials.

More recently, the connection between RBF interpolation, *Generalized Least Squares (GLSQ)* [49] and its extension to the Best Unbiased Linear Estimator (BLUE) problem has been established [39, 33, 32]. If the covariance matrix of a GLSQ (and BLUE) problem is described by a symmetric kernel matrix of an RBF problem among other conditions, the two problems become equivalent. Although GLSQ is of high interest to the statistics community, as shown by the high number of citations of [49], the lack of fast solvers limits its application to small to medium size problems [33, 32]. Moreover, many of these statistical problems involve higher than zero- and linear-order polynomial regression [57, 58, 47, 34, 35, 54]. By exploiting the connection between GLSQ and RBFs, we will be able to solve GLSQ using the fast solvers developed in the RBF and integral equation communities.

For the BLUE Kriging estimator there is less need of higher order polynomials. In many cases quadratic is sufficient for high accuracy estimation. For the matlab DACE Kriging toolbox [33, 32] the highest polynomial accuracy is set to quadratic. In our paper we demonstrate the difference between constant, linear and quadratic BLUE estimates. As shown in the results in Section 4, the quadratic interpolant leads to much better estimate than constant and linear ones. In addition, in [30] the author uses second order polynomial BLUE for repairing surfaces.

Recently Gumerov et al. [27] developed a RBF solver with a Krylov subspace method in conjunction with a preconditioner constructed from Cardinal functions. We note that this approach, to our knowledge, is the state of the art for zero-order interpolation in $\mathbb{R}^3$ with a biharmonic spline. This makes it very useful for interpolation problems in computer graphics. On the other hand, its application to regression problems such as GLSQ is rather limited.

A domain decomposition method was developed in [8] by Beatson et al. This method is a modification of the Von Neumann's alternating algorithm, where the global solution is approximated by iterating a series of local RBF interpolation problems. This method is promising and has led to (coupled with multi-pole expansions) $O(N \log(N))$ computational cost for certain interpolation problems.

Although the method is very efficient and exhibits $O(N \log(N))$ computational complexity, this seems to be true for small to medium size problems (up to 50,000 nodes in $\mathbb{R}^3$) with smooth data. Beyond that range the computational cost increases quadratically as shown in [8]. Other results for non smooth data shows that the computational complexity is more erratic [15]. Furthermore, in many cases, it is not obvious how to pick the optimal domain decomposition scheme.

An alternative approach was developed by Beatson et al. [7], which is based on preconditioning and coupled with GMRES iterations [48]. This approach relies on the construction of a polynomial orthogonal basis, similar to the HB approach in our paper. This approach gives rise to a highly sparse representation of the RBF interpolation matrix that can be very easily preconditioned by means of a diagonal matrix. The new system of equations exhibits condition number growth of no more than $\mathcal{O}(\log N)$. The downside is that this basis is not complete. This is ameliorated by the introduction of non decaying elements, but no guarantees on accuracy can be made.

Our approach is based on posing the RBF interpolation problem as a discretization of an integral equation and applying preconditioning techniques. This approach has many parallels with the work developed by Beatson et al. [7]. However, our approach was developed from work done for fast integral equation solvers.

Most of the work in the area of fast integral equation solvers has been restricted to the efficient computation of matrix vector products as part of an iterative scheme. For the Poisson kernel the much celebrated multi-pole spherical harmonic expansions leads to a fast summation algorithm that reduces each matrix-vector multiplication to $O(N)$ computational steps [26, 6]. This technique has been extended to a class of polyharmonic splines and multiquadrics [5, 20]. More recently L. Ying et al. has developed multipole algorithms for a general class of kernels [59]. In contrast, the development of optimal (or good) preconditioners for integral equations has been more limited.

A unified approach for solving integral equations efficiently was introduced in [1, 2, 9]. A wavelet basis was used for sparsifying the discretized operator and only $O(N \log_2^2(N))$ entries of the discretization matrix are needed to achieve optimal asymptotic convergence. The downside is that it was limited to 1D problems.

In [18] a class of multiwavelets based on a generalization of Hierarchical Basis (HB) functions was introduced for sparsifying integral equations on conformal surface meshes in $\mathbb{R}^3$. These wavelets are continuous, multi-dimensional, multi-resolution and spatially adaptive. These constructions are based on the work on *Lifting* by Schroder and Sweldens [52] and lead to a class of adapted HB of arbitrary polynomial order. A similar approach was also developed in [55].

These constructions provide compression capabilities that are independent of the geometry and require only $O(N \log_4^{3.5}(N))$ entries to achieve optimal asymptotic convergence. This is also true for complex geometrical features with sharp edges. Moreover, this basis has a multi-resolution structure that is related to the BPX scheme [42], making them an excellent basis to precondition integral and partial differential equations. In [28] Heedene et al. demonstrate how to use this basis to build scale decoupled stiffness matrices for partial differential equations (PDEs) over non uniform irregular conformal meshes.

In this paper, we develop a discrete HB for solving isotropic RBF interpolation problems efficiently. Our HB construction is adapted to the topology of the interpolating nodes and the kernel. This new basis decouples the polynomial interpolation from the RBF part, leading to a system of equations that are easier to solve. With our sparse SSOR [25, 31] or diagonal preconditioner, combined with a fast summation method, the RBF interpolation problem can be solved efficiently.

Our contributions include a method with complexity costs similar to Gumerov et al [27] for problems in $\mathbb{R}^3$, albeit for the biharmonic spline with constant order their approach shows efficiency superior to ours. However, due to the decoupling of the polynomial interpolation, our approach is more flexible and works well for higher order polynomials. We show similar results for the multiquadric RBFs in $\mathbb{R}^3$ (we did not observe multiquadric results for $\mathbb{R}^3$ in [27] and to our knowledge we have not found any). Note that the idea of decoupling the RBF system of equations from the polynomial interpolation has also been proposed in [53] and [8].

In the rest of Section 1 we explicitly pose the RBF interpolation problem. In Section 2, we construct an HB that is adapted to the interpolating nodes and the kernel seed function. In Section 3 we demonstrate how the adapted HB is used to form a multi-resolution RBF matrix, which is used to solve the interpolation problem

efficiently. In section 4, we show some numerical results of our method. The interpolating nodes are randomly placed, moreover the interpolating values themselves contain random noise. We summarize our conclusions in section 5.

After we submitted this paper we became aware of the H-Matrix approach by Hackbusch [11] applied to stochastic capacitance extraction [61] problem. In [10] the authors apply an H-matrix approach to sparsify the kernel matrix arising from a Gaussian process regression problem to $\mathcal{O}(N \log N)$. In our paper, we apply HB to precondition the RBF system, although we could also use them to sparsify it. Instead, we use a fast summation approach to compute the matrix-vector products.

There are many similarities between our approach and the H-matrix method. We shall discuss this topic more in detail in the conclusion in Section 5.

1.1. Radial Basis Function Interpolation. In this section we pose the problem of RBF interpolation for bounded functions defined on $\mathbb{R}^3$. Although our exposition is only for $\mathbb{R}^3$, the RBF problem and our HB approach can be easily extended to any finite dimensions.

Consider a function $f(\vec{x}) : \mathbb{R}^3 \to \mathbb{R}$ and its evaluation on a set of user-specified sampling of distinct nodes $X := \{\vec{x}_1, ..., \vec{x}_N\} \subset \mathbb{R}^3$, where $\vec{x} = [x_1, x_2, x_3]^H$, unisolvent with respect to all polynomials of degree at most m. We are interested in constructing approximations to $f(\vec{x})$ of the form

$$s(\vec{x}) = \sum_{i=1}^{M(m)} c[i] q_i(\vec{x}) + \sum_{j=1}^{N} u[j] K(\vec{x}, \vec{x}_j),$$

where $K : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$, $u \in \mathbb{R}^N, c \in \mathbb{R}^{M(m)}$ and $P := \{q_1(\vec{x}), \ldots, q_{M(m)}(\vec{x})\}$ is a basis for $\mathcal{P}^m(\mathbb{R}^3)$, i.e. the set of all polynomials of total degree at most m in $\mathbb{R}^3$ (Note that $M(m)$ is the number of polynomials that form a basis for $\mathcal{P}^m(\mathbb{R}^3)$ i.e. $M(m) = \binom{m+3}{3}$). This interpolant must satisfy the following condition

$$s(\vec{x}_j) = f(\vec{x}_j), \quad j = 1, \ldots, N,$$

for all $\vec{x}_j$ in X. Moreover, to ensure the interpolation is unique we add the following constraint

$$\sum_{j=1}^{N} u[j] q\left(\vec{x}_j\right) = 0, \tag{1.1}$$

for all polynomials $q(\vec{x})$ of degree at most m. Now, since $M(m)$ is the minimum amount of nodes needed to solve the polynomial problem, we need at least $N \geqslant M(m)$ RBF centers. The interpolation problem can be rewritten in matrix format as

$$\begin{pmatrix} K & Q \\ Q^H & O \end{pmatrix} \begin{pmatrix} u \\ c \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}, \tag{1.2}$$

where $K_{i,j} = K(\vec{x}_i, \vec{x}_j)$ with i, j running from 1 to n; $d \in \mathbb{R}^N$ such that $d_j = f(\vec{x}_j)$; $c \in \mathbb{R}^{M(m)}$; and $Q_{i,j} = q_j(\vec{x}_i)$ with i running from 1 to N, j runs from 1 to $M(m)$. Denote the columns of Q as $[q_1, \ldots, q_{M(m)}]$.

This is the general form of the RBF interpolation isotropic problem. The properties of this approximation mostly depend on the seed function $K(\vec{x}, \vec{y})$. An example of a well known isotropic kernel in $\mathbb{R}^3$ is the biharmonic spline

$$K(\vec{x}, \vec{x}_j) := K(|\vec{x} - \vec{x}_j|) = |\vec{x} - \vec{x}_j|\,. \tag{1.3}$$

This is a popular kernel due to the optimal smoothness of the interpolant [8]. This kernel has been successfully applied in point cloud reconstructions, denoising and repairing of meshes [14]. More recently, there has been interest in extensions to anisotropic kernels [16, 17], i.e.

$$K(\vec{x}, \vec{x}_j) := K\left(|T_j(\vec{x} - \vec{x}_j)|\right),$$

where T_j is a 3×3 matrix. The stabilization method introduced in this paper can be extended to solving efficiently the RBF problem with spatially varying kernels. By using the sparsification properties of the adapted HB a sparse representation of the spatially varying RBF matrix can be constructed in optimal time. However, in this paper we restrict the analysis to isotropic kernels in $\mathbb{R}^3$, i.e. $T_j = \alpha I$ where $\alpha > 0$.

One aspect of RBF interpolation is the invertibility of the matrix in Equation (1.2). In [37] it is shown that the interpolation problem (1.2) has a unique solution if we assume that the interpolating nodes in X are unisolvent with respect to $\mathcal{P}^m(\mathbb{R}^3)$ and the continuous kernel is *strictly conditionally positive (or negative) definite*. Before we give the definition, we provide some notation.

DEFINITION 1.1. *Suppose that $X \subset \mathbb{R}^3$ is a set of interpolating nodes and $\{q_1(\vec{x}), q_2(\vec{x}), \ldots, q_{M(m)}(\vec{x})\}$ is a basis for $\mathcal{P}^m(\mathbb{R}^3)$, then we use $\mathcal{P}^m(X)$ to denote the column space of Q.*

We now assume the kernel matrix K satisfies the following assumption.

DEFINITION 1.2. *We say that the symmetric function $K(\cdot, \cdot) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is strictly conditionally positive definite of order l if for all sets $X \subset \mathbb{R}^3$ of distinct nodes*

$$v^H K v = \sum_{i,j}^{N} v_i v_j K(\vec{x}_i, \vec{x}_j) > 0,$$

for all $v \in \mathbb{R}^N$ such that $v \perp \mathcal{P}^l(X)$ and $v \neq 0$. Alternatively, under the same assumptions, $K(\cdot, \cdot) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is strictly conditionally negative definite if

$$v^H K v < 0,$$

for all $v \in \mathbb{R}^N$ such that $v \perp \mathcal{P}^l(X)$.

REMARK 1. *Many Kernels, including biharmonics and multiquadrics, satisfy the strictly conditionally positive or negative assumption. By a theorem due to Micchelli [37], this assumption is satisfied for all continuous kernels with completely monotonic derivative on $(0, \infty)$.*

The invertibility of the RBF interpolation problem can be proven by the basis construction developed in this paper. Although this is not necessary, it does cast insights on how to construct a basis that can solve the RBF Problem (1.2) efficiently.

1.2. Decoupling of the RBF interpolation problem. Suppose there exists a matrix $T : \mathbb{R}^{N-M} \to \mathbb{R}^N$, where $M := dim(\mathcal{P}^m(X))$, such that T^H annihilates any vector $v \in \mathcal{P}^m(X)$ (i.e. $T^H v = 0 \quad \forall v \in \mathcal{P}^m(X)$). Furthermore, suppose there exists a second matrix $L : \mathbb{R}^M \to \mathbb{R}^N$ such that the combined matrix $P := [L \;\; T]$ is orthonormal such that $P^H : \mathbb{R}^N \to \mathbb{R}^N$ maps $\mathbb{R}^N$ onto

$$\mathcal{P}^m(X) \oplus W,$$

where $W := (\mathcal{P}^m(X))^\perp$.

Suppose that $u \in \mathcal{P}^m(X)^\perp$, then $u = Tw$ for some $w \in \mathbb{R}^{N-M}$. Problem (1.2) can now be re-written as

$$T^H K T w + T^H Q c = T^H d.$$

However, since the columns of Q belong in $\mathcal{P}^m(X)$ then

$$T^H K T w = T^H d. \tag{1.4}$$

From Definition 1.2 and the orthonormality of P we conclude that w can be solved uniquely. The second step is to solve the equation $L^H Q c = L^H d - L^H K T w$. From the unisolvent property of the nodes X the matrix Q has rank $dim(\mathcal{P}^m(\mathbb{R}^3))$, moreover, L also has rank $dim(\mathcal{P}^m(\mathbb{R}^3))$, thus $L^H Q$ has full rank and it is invertible.

Although proving the existence of P and hence the uniqueness of the RBF problem is an interesting exercise, there are more practical implications to the construction of P. First, the coupling of Q and K can lead to a system of ill-conditioned equations depending on the scale of the domain [8]. The decoupling property of the transform P leads to a scale independent problem, thus correcting this source of ill-conditioning. But more importantly, we focus on the structure of $T^H K T$ and how to exploit it to solve the RBF interpolation problem (1.2) efficiently. The key idea is the ability of T^H to vanish discrete polynomial moments and its effect on the matrix $K(\cdot,\cdot)$. We shall now restrict our attention to Kernels that satisfy the following assumption.

ASSUMPTION 1. *Let* $D_x^\alpha := \frac{\partial^{\alpha_1,\alpha_2,\alpha_3}}{\partial \vec{x}_1^{\alpha_1} \partial \vec{x}_2^{\alpha_2} \partial \vec{x}_3^{\alpha_3}}$ *and similarly for* D_y^β*, we assume that*

$$D_x^\alpha D_y^\beta K(\vec{x}, \vec{y}) \leqslant \frac{C}{|\vec{x} - \vec{y}|^{q+|\alpha|+|\beta|}},$$

where $\alpha = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{Z}^3$, $|\alpha| = \alpha_1 + \alpha_2 + \alpha_3$, *and* $q \in \mathbb{Z}$. *In addition, we assume that* $K(\vec{x}, \cdot)$ *and* $K(\cdot, \vec{y})$ *are analytic everywhere except for* $\vec{x} = \vec{y}$. This assumption is satisfied by many practical kernels, such as multiquadrics and polyharmonic splines [24, 8].

2. Adapted Discrete Hierarchical Basis Constructions. In this section we show how to construct a class of discrete HB that is adapted to the kernel function $K(\cdot,\cdot)$ and to the local interpolating nodes (or interpolating nodes) contained in X. The objective is to solve RBF interpolation Problem (1.2) efficiently. The HB method will be divided into the following parts:

- **Multi-resolution domain decomposition.** The first part is in essence a pre-processing step to build cubes at different levels of resolution as place holders for the interpolation nodes belonging to X.

- **Adapted discrete HB construction.** From the multi-resolution domain decomposition of the interpolating nodes in X, an adapted multi-resolution basis is constructed that annihilates any polynomial in $\mathcal{P}^p(X)$, where $p \in \mathbb{Z}^+$ and $p \geqslant m$. p will be in essense the order of the Hierarchical Basis, which is not to be confused with m, which is the order of the RBF problem.
- **GMRES iterations with fast summation method.** With the adapted HB a multi-resolution RBF interpolation matrix is implicitly obtained through a fast summation method and solved iteratively with a GMRES algorithm and an SSOR or diagonal preconditioner.

2.1. Multi-resolution Domain Decomposition. The first step in constructing an adapted HB is to rescale and embed the interpolating nodes in X into a cube $B_0^0 := [-0.5, 0.5]^3$. The next step is to form a series of level dependent cubes that serve as place holders for the interpolating nodes at each level of resolution.

The basic algorithm is to subdivide the cube B_0^0 into eight cubes if $|B_0^0| > M(p)$, where $|B_k^j|$ denotes the total number of interpolating nodes contained in the cube B_j^k. Subsequently, each cube B_k^j is sub-divided if $|B_k^j| > M(p)$ until there are at most $M(p)$ interpolating nodes at the finest level. The algorithm is explained more in detail in the following pseudo-code:

Input: $X := \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$, $M(p)$
Output: $B_k^j \ \forall k \in \{\mathcal{K}(0), \ldots, \mathcal{K}(n)\}, n$
begin
  pre-processing;
  $q \leftarrow \mathbf{max}_{\vec{x}_i, \vec{x}_j \in X} \|\vec{x}_i - \vec{x}_j\|$;
  forall the $\vec{x}_i \in X$ **do**
    $\vec{c} \leftarrow \frac{\sum_{i=1}^{N} \vec{x}_i}{N}$;
  end
  for $i \leftarrow 1$ **to** N **do**
    $\vec{x}_i \leftarrow \frac{\vec{x}_i - \vec{c}}{q}$;
  end
  $j \leftarrow 0$;
  $B_0^0 \leftarrow [-0.5, 0.5]^3$;
  $\mathcal{K}(0) \leftarrow \{0\}$;
  main;
  while $|B_k^j| > M(p)$ **for any** $k \in \mathcal{K}(j)$ **do**
    $\mathcal{K}(j+1) \leftarrow \emptyset$;
    for $k \leftarrow 0$ **to** $|\mathcal{K}(j)|$ **do**
      if $|B_k^j| > M(p)$ **then**
        subdivide into eight cubes $\{B_{8k}^{j+1}, \ldots, B_{8k+7}^{j+1}\}$;
        $\mathcal{K}(j+1) \leftarrow \mathcal{K}(j+1) \cup_{w=0}^{7} 8k + w$;
      end
    end
    $j \leftarrow j + 1$;
  end
  $n \leftarrow j$
end

Algorithm 1: Multi-resolution Domain Decomposition

REMARK 2. *$\mathcal{K}(j)$ is an index set for all the cubes at level j. We use $|\mathcal{K}(j)|$ to*

denote the cardinality of $\mathcal{K}(j)$.

REMARK 3. *Finding the maximal distance between any two nodes can be performed in* $\mathcal{O}(N(n+1))$ *computational steps by applying an octree algorithm. Therefore the Multi-resolution Domain Decomposition algorithm can be performed in* $\mathcal{O}(N(n+1))$ *computational steps. This can be easily seen since the maximum number of boxes at any level* j *is bounded by* N *and there is a total of* $n+1$ *levels.*

Before describing the construction of the adapted discrete HB, we introduce some more notations to facilitate our discussion.

DEFINITION 2.1. *Let* $\mathcal{B}_j$ *be the set of all the cubes* B_k^j *at level* j *that contain at least one interpolating center from* X.

DEFINITION 2.2. *Let* $\mathbf{C} := \{e_1, \ldots, e_N\}$, *where* $e_i[i] = 1$ *and* $e_i[j] = 0$ *if* $i \neq j$. *Furthermore, define the bijective mapping* $F_p : \mathbf{C} \to X$ *such that* $F_p(e_i) = \vec{x}_i$, *for* $i = 1 \ldots N$ *and* $F_q : \mathbf{C} \to Z^+$ *s.t.* $F_q(e_i) = i$. *Now, for each cube* $B_k^n \in \mathcal{B}_n$ *at the finest level* n, *let*

$$\mathbf{B}_k^n := \{e_i \mid F_p(e_i) \in B_k^n\}.$$

DEFINITION 2.3. *Let* $\mathcal{C}_n := \bigcup_{k \in \mathcal{K}(n)} \mathbf{B}_k^n$.

DEFINITION 2.4. *Let children*(B_k^j) *be the set of nonempty subdivided cubes* $\{B_l^{j+1}, \ldots, B_{l+w}^{j+1}\} \in \mathcal{B}_j$ *of* B_k^j.

DEFINITION 2.5. *For every non empty* B_k^j *let the set parent*$(B_k^j) := \{B_l^{j-1} \in \mathcal{B}_{j-1} \mid B_k^j \in children(B_l^{j-1})\}$.

2.2. Basis Construction. From the output of the multi-resolution decomposition Algorithm 1 we can now build an adapted discrete HB that annihilates any polynomial in $\mathcal{P}^p(X)$. To construct such a basis, we apply the *stable completion* [13] procedure. This approach was followed in [18]. However, the basis is further orthogonalized by using a modified Singular Value Decomposition (SVD) orthonormalization approach introduced in [55].

Suppose $v_1, \ldots, v_s$ are a set of orthonormal vectors in $\mathbb{R}^N$, where $s \in \mathbb{Z}^+$, a new basis is constructed such that

$$\phi_j := \sum_{i=1}^{s} c_{i,j} v_i, \;\; j = 1, \ldots, a; \qquad \psi_j := \sum_{i=1}^{s} d_{i,j} v_i, \;\; j = a+1, \ldots, s,$$

where $c_{i,j}$, $d_{i,j} \in \mathbb{R}$ and for some $a \in \mathbb{Z}^+$. We desire that the new discrete HB vector ψ_j to be orthogonal to $\mathcal{P}^p(X)$, i.e.

$$\sum_{k=1}^{N} r[k]\psi_j[k] = 0, \tag{2.1}$$

for all $r \in \mathcal{P}^p(X)$. Notice that the summation and the vectors r and ψ_j are in the same order as the entries of the set X.

Due to the orthonormality of the basis $\{v_i\}_{i=1}^s$ this implies that Equation (2.1) is satisfied if the vector $[d_{i,1}, \ldots, d_{i,s}]$ belongs to the null space of the matrix

$$M_{s,p} := Q^H V,$$

where the columns of Q are a basis for $\mathcal{P}^p(X)$ (i.e. all the polynomial moments) and $V = [v_1, v_2, \ldots, v_s]$. (Notice that the order of the summation is done with respect to

the set X). Suppose that the matrix $M_{s,p}$ is a rank a matrix and let $U_{s,p}D_{s,p}V_{s,p}$ be the SVD decomposition. We then pick

$$\left[\begin{array}{ccc|ccc} c_{0,1} & \dots & c_{a,1} & d_{a+1,1} & \dots & d_{s,1} \\ c_{0,2} & \dots & c_{a,2} & d_{a+1,2} & \dots & d_{s,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{0,s} & \dots & c_{a,s} & d_{a+1,s} & \dots & d_{s,s} \end{array}\right] := V_{s,p}^H, \tag{2.2}$$

where the columns $a+1, \dots, s$ form an orthonormal basis of the nullspace $\mathcal{N}(M_{s,p})$. Similarly, the columns $1, \dots, a$ form an orthonormal basis of $\mathbb{R}^N \backslash \mathcal{N}(M_{s,p})$.

REMARK 4. *If $\{v_1, \dots, v_s\}$ is orthonormal, then new basis $\{\phi_1, \dots, \phi_a, \psi_{a+1}, \dots, \psi_s\}$ is orthonormal, and spans the same space as span$\{v_1, \dots, v_s\}$. This is due to the orthonormality of the matrix $V_{s,p}$.*

REMARK 5. *If s is larger than the total number of vanishing moments, then $M_{s,p}$ is guaranteed to have a nullspace of at least rank $s - M(p)$, i.e. there exist at least $s - M(p)$ orthonormal vectors $\{\psi_i\}$ that satisfy Equation 2.1.*

2.3. Finest Level. We can now build an orthonormal multi-resolution basis. First, choose a priori the order of moments p and start at the highest level resolution level n. The next step is to progressively build the adapted HB as the levels are traversed.

At the finest level n, for each cube $\mathbf{B}_k^n \in \mathcal{C}_n$ let $v_i := e_i$ for all $e_i \in \mathbf{B}_k^n$. As described in the previous section, the objective is to build new functions

$$\phi_{k,l}^n := \sum_{i=1}^{s} c_{n,i,l,k} v_i, \;\; l = 1, \dots, a_{n,k}, \qquad \psi_{k,l}^n := \sum_{i=1}^{s} d_{n,i,l,k} v_i, \;\; l = a_{n,k}+1, \dots, s,$$

such that Equation (2.1) is satisfied.

The first step is to form the matrix $M_{s,p}^{n,k} := Q^H V$, where the columns of Q are a basis for $\mathcal{P}^p(X)$. Notice that since $e_i[w] = 0$ for $w \neq i$ and $e_i \in \mathbf{B}_k^n$, then only $|B_k^n|$ columns of Q^H are need to form the matrix $M_{s,p}^{n,k}$ and the rest can be thrown away since they multiply with zero.

The next step is to apply the SVD procedure such that $M_{s,p}^{n,k} \to U_{s,p}^{n,k} D_{s,p}^{n,k} V_{s,p}^{n,k}$. The coefficients $c_{n,i,j,k}$ and $d_{n,i,j,k}$ are then obtained from the rows of $V_{s,p}^{n,k}$ and $a_{n,k} := \operatorname{rank} M_{s,p}$.

Now, for each $B_k^n \in \mathcal{B}^n$ denote $\bar{C}_k^n$ as the collection of basis vectors $\{\phi_{k,1}^n, \dots, \phi_{k,a_{n,k}}^n\}$, and similarly denote $\bar{D}_k^n$ as the collection of basis vectors $\{\psi_{k,a_{n,k}+1}^n, \dots, \psi_{k,s}^n\}$. Furthermore, we define the *detail* subspace

$$W_k^n := span\{\psi_{k,a_{n,k}+1}^n, \dots, \psi_{k,s}^n\}$$

and the *average* subspace

$$V_k^n := span\{\phi_{k,1}^n, \dots, \phi_{k,a_{n,k}}^n\}.$$

By collecting the transformed vectors from all the cubes in $\mathcal{C}_n$, we form the subspaces

$$V^n := \oplus_{k \in \mathcal{K}(n)} V^{n,k}, \;\; W^n := \oplus_{k \in \mathcal{K}(n)} W^{n,k},$$

where $\oplus$ is a direct sum and $\mathcal{K}(i) := \{k \mid B_k^i \in \mathcal{B}^i\}$.

REMARK 6. *We first observe that $\mathbb{R}^N = V^n \oplus W^n$. This is true since the number of interpolating nodes is equal to N and $\mathbb{R}^N = span\{e_1, e_2, \ldots, e_N\}$.*

REMARK 7. *It is possible that $W_k^n = \emptyset$ for some particular cube B_k^n. This will be the case if the cardinality of $\mathbf{B}_k^n$ is less or equal to $M(p)$ i.e. the dimension of the nullspace of $M_{s,p}$ is zero. However, this will not be a problem. As we shall see in section 2.4, the next set of HB are built from the vectors in $\bar{C}_k^n$ and its siblings.*

LEMMA 2.6. *The basis vectors of V^n and W^n form an orthonormal set.*

Proof. First notice that since $\mathbf{B}_l^n \cap \mathbf{B}_k^n = \emptyset$ whenever $k \neq l$ then $V^{n,k} \perp V^{n,l}$, $W^{n,k} \perp W^{n,l}$ and $V^{n,k} \perp W^{n,l}$. The result follows from the fact that the rows $V_{s,p}^{n,k}$ form an orthonormal set. ∎

It is clear that the *detail* subspace $W^n \perp \mathcal{P}^p(\vec{x})$, but the *average* subspace V^n is not. However, we can still perform the SVD procedure to further decompose V^n. To this end we need to accumulate the *average* basis vectors of V^n. For each $B_k^{n-1} \in \mathcal{B}_{n-1}$ identify the set $children(B_k^{n-1})$. Form the set $\mathbf{B}_k^{n-1} := \{\bar{C}_l^n \mid B_l^n \in children(B_k^{n-1})\}$. We can now apply the SVD procedure on each set of average vectors in $\mathbf{B}_k^{n-1}$.

2.4. Intermediate Level. Suppose we have the collection of sets $\mathbf{B}_k^i$ for all $k \in \mathcal{K}(i)$. For each $\mathbf{B}_k^i$ perform the matrix decomposition $M_{s,p}^{i,k} = U_{s,p}^{i,k} D_{s,p}^{i,k} V_{s,p}^{i,k}$ for all $v \in \mathbf{B}_k^i$. From the matrix $V_{s,p}^{i,k}$ obtain the decomposition

$$\phi_{k,l}^i := \sum_{j=1}^{s} c_{i,j,l,k} v_j, \;\; l = 1, \ldots, a_{i,k}, \qquad \psi_{k,l}^i := \sum_{j=1}^{s} d_{i,j,l,k} v_j, \;\; l = a_{i,k}+1, \ldots, s,$$

where the coefficients $c_{i,j,l,k}$ and $d_{i,j,l,k}$ are obtained from the rows of $V_{s,p}^{i,k}$ in (2.2) and $a_{i,k} := \operatorname{rank} M_{s,p}^{i,k}$. Then we form the subspaces

$$W_k^i := span\{\psi_{k,a_{i,k}+1}^i, \ldots, \psi_{k,s}^i\}, \;\; V_k^i := span\{\phi_{k,1}^i, \ldots, \phi_{k,a_{i,k}}^i\},$$

and

$$V^i := \oplus_{k \in \mathcal{K}(i)} V^{i,k}, \;\; W^i := \oplus_{k \in \mathcal{K}(i)} W^{i,k}.$$

It is easy to see that $V^{i+1} = V^i \oplus W^i$. The basis vectors are collected into two groups:

DEFINITION 2.7. *For each $B_k^i \in \mathcal{B}^i$ let the sets $\bar{C}_k^i := \{\phi_{k,1}^i, \ldots, \phi_{k,a_{i,k}}^i\}$, and $\bar{D}_k^i := \{\psi_{k,a_{n,k}+1}^i, \ldots, \psi_{k,s}^i\}$.*

Just as for the finest level case, we can further decompose V^i. To this end, for each $B_k^{i-1} \in \mathcal{B}_{i-1}$ identify the set $children(B_k^{i-1})$. Form the set $\mathbf{B}_k^{i-1} := \{\bar{C}_l^i \mid B_l^i \in children(B_k^{i-1})\}$.

2.5. Coarsest Level. It is clear that when the iteration reaches V^0 the basis function no longer annihilates polynomial moments of order p. However, a new basis can be obtained that can vanish polynomial moments $m \leqslant p$.

Recall that for the RBF interpolation problem with polynomial order m it is imposed that $u \perp \mathcal{P}^m(X)$. If $p = m$ then it is clear that $u \in W^0 \oplus \ldots W^n$ and RBF problem decouples as shown in Section 1. However, if $p > m$ then $u \in \mathcal{P}^p(X) \backslash \mathcal{P}^m(X) \oplus W^0 \oplus \ldots W^n$ and the RBF problem does not decouple. It is then of interest to find an orthonormal basis to

$$\mathcal{P}^p(X) \backslash \mathcal{P}^m(X).$$

This can be easily achieved. Let the columns of the matrix

$$\left[\begin{bmatrix} q_1(\vec{x}_1) \\ \vdots \\ q_1(\vec{x}_N) \end{bmatrix}, \begin{bmatrix} q_2(\vec{x}_1) \\ \vdots \\ q_2(\vec{x}_N) \end{bmatrix}, \ldots, \begin{bmatrix} q_{\tilde{m}}(\vec{x}_1) \\ \vdots \\ q_{\tilde{m}}(\vec{x}_N) \end{bmatrix}, \ldots, \begin{bmatrix} q_{\tilde{p}}(\vec{x}_1) \\ \vdots \\ q_{\tilde{p}}(\vec{x}_N) \end{bmatrix}\right]$$

be a basis for $\mathcal{P}^p(X)$, where each function $q_i(x)$ corresponds to the i^{th} moment. Now, the first $M(m)$ columns correspond to a basis for $\mathcal{P}^m(X)$. Thus an orthonormal basis for $\mathcal{P}^p(X)\backslash\mathcal{P}^m(X)$ is easily achieved by applying the Gram-Schmidt process.

Alternatively the matrix $M_{0,m}$ can now be formed by applying the SVD decomposition and a basis that annihilates all polynomial of order m or lower is obtained. The matrix $\bar{C}_0^0$ can now be replaced with the matrix $[C_0^{-1}, D_0^{-1}]$, where the columns of C_0^{-1} form an orthonormal basis for $\mathcal{P}^m(X)$ and D_0^{-1} is an orthonormal basis for $\mathcal{P}^p(X)\backslash\mathcal{P}^m(X)$.

The complete algorithm to decompose $\mathbb{R}^N$ into a multi-resolution basis with respect to the interpolating nodes X is described in Algorithms 2 and 3.

Input: Finest level n; Order of RBF m; $B_k^j \ \forall k \in \mathcal{K}(j), j = -1 \ldots n$;
$\mathbf{B}_k^n \ \forall k \in \mathcal{K}(n)$; Degree of vanishing moments $p \geqslant m$; X.
Output: $\{\bar{C}_0^{-1}, \bar{D}_0^{-1}, \bar{D}_0^0, \bar{D}_1^0, \ldots, \bar{D}_k^n\}$
main;
for $j \leftarrow n$ **to** 1 **step** -1 **do**
  for $k \leftarrow 1$ **to** $|\mathcal{K}(j-1)|$ **do**
    $\mathbf{B}_k^{j-1} \leftarrow \emptyset$
  end
  for $k \leftarrow 1$ **to** $|\mathcal{K}(j)|$ **do**
    $\{\bar{D}_k^j, \bar{C}_k^j\} \leftarrow$ PolyOrtho($\mathbf{B}_k^j$, p);
    $U \leftarrow parent(B_k^j)$;
    forall the $B_l^{j-1} \in U$ **do**
      $\mathbf{B}_l^{j-1} \leftarrow \mathbf{B}_l^{j-1} \cup \bar{C}_k^j$
    end
  end
end
$\{\bar{D}_0^{-1}, \bar{C}_0^{-1}\} \leftarrow$ PolyOrtho($\mathbf{B}_0^0$,m);

Algorithm 2: Adapted Discrete HB Construction

LEMMA 2.8.

$$\mathbb{R}^N = V^0 \oplus W^0 \oplus \ldots W^n = span\{\bar{C}_0^{-1}, \bar{D}_0^{-1}, \bar{D}_0^0, \bar{D}_1^0, \ldots, \bar{D}_k^n\}.$$

for $j = 0 \ldots n$ and for all $k \in \mathcal{K}(j)$

Proof. The result follows from Remark 6 and that V_i is decomposed into $V^{i-1} \oplus W^{i-1}$ for all $i = 1 \ldots n$. ∎

REMARK 8. *When Algorithm 2 terminates at level $i = 0$, there will be $M(p)$ orthonormal vectors that span $\mathcal{P}^p(X)$.*

REMARK 9. *At the finest level n, the number of vectors in each matrix $\bar{C}_k^n$ corresponding to $\mathbf{B}_k^n$ is bounded by $M(p)$. Now, for each $\mathbf{B}_k^{n-1}$ there are at most $8M(p)$ vectors from the children of $\mathbf{B}_k^{n-1}$. From the procedure for the basis construction in section 2.2 for each $\mathbf{B}_k^{n-1}$ there are at at most $M(p)$ vectors in $\bar{C}_k^n$. Furthermore,*

there are no more than $8M(p)$ vectors in $\bar{D}_k^n$ formed. The same conclusion follows for each $\mathbf{B}_k^i$, for all levels $i = 0, \ldots, n$.

Input: $\mathbf{B}_k^j$, Degree of vanishing moment p
Output: $\bar{D}_k^j$, $\bar{C}_k^j$
$V \leftarrow \emptyset$;
forall the $v_i \in \mathbf{B}_k^j$ **do**
  $V \leftarrow [V, v_i]$
end
Form the $N \times M(p)$ polynomial basis matrix Q;
$M_{s,p}^{j,k} \leftarrow Q^H V$;
$[U_{s,p}^{j,k}, D_{s,p}^{j,k}, V_{s,p}^{j,k}] \leftarrow SVD(M_{s,p}^{j,k})$;
$s \leftarrow |\mathbf{B}_k^j|$;
$a_{j,k} \leftarrow$ rank of $D_{s,p}^{j,k}$;
for $l \leftarrow 1$ **to** $a_{j,k}$ **do**
  $\phi_{k,l}^j \leftarrow 0$;
  for $i \leftarrow 1$ **to** s **do**
    $\phi_{k,l}^j \leftarrow \phi_{k,l}^j + c_{j,i,l,k} v_i$
  end
  $\bar{C}_k^i \leftarrow [\bar{C}_k^j,\ \phi_{k,l}^j]$
end
for $l \leftarrow a_{j,k} + 1$ **to** s **do**
  $\psi_{k,l}^j \leftarrow 0$;
  for $i \leftarrow 1$ **to** s **do**
    $\psi_{k,l}^j \leftarrow \psi_{k,l}^j + d_{j,i,l,k} v_i$
  end
  $\bar{D}_k^j \leftarrow [\bar{D}_k^j, \psi_{k,l}^j]$
end

Algorithm 3: PolyOrtho($\mathbf{B}_k^j$,p)

DEFINITION 2.9. *For any $\mathbf{B}_k^i$, $k \in \mathcal{K}(i)$, $i = 0, \ldots n$, let $|\mathbf{B}_k^i|$ be the number of vectors in $\mathbf{B}_k^i$.*

THEOREM 2.10. *The complexity cost for Algorithm 2 is bounded by $\mathcal{O}(N(n+1))$.*

Proof. Suppose we start at the finest level n. Now, for each box in B_k^n, the vectors $e_i \in \mathbf{B}_k^n$ have at most one non-zero entry. This implies that the matrix $M_{s,p}^{n,k} = Q^H V$, Q^H is a $M(p) \times |B_k^n|$ matrix and V is at most a $|B_k^n| \times |B_k^n|$ matrix. Then the total cost to computing $M_{s,p}^{n,k}$ for all $k \in \mathcal{K}(n)$ is bounded by

$$C \sum_{k \in \mathcal{K}(n)} |B_k^n|^2 M(p)$$

for some $C > 0$. Now since $\left|\cup_{k \in \mathcal{K}(n)} B_k^n\right| = N$ and $|B_k^n|$ is at most $M(p)$ $\forall k \in \mathcal{K}(n)$, then the cost for computing $\bar{C}_k^n$ and $\bar{D}_k^n$, $\forall k \in \mathcal{K}(n)$, is at most $\mathcal{O}(N)$.

At level $n-1$, from Remark 9 we see that there are at most $8M(p)$ vectors in each $\mathbf{B}_k^{n-1}$ $\forall k \in \mathcal{K}(n-1)$. Forming the the matrix $M_{s,p}^{n-1,k} = Q^H V$, Q^H is at most $M(p) \times |B_k^{n-1}|$ and V is at most $|B_k^{n-1}| \times |\mathbf{B}_k^{n-1}|$. Now, since $\left|\cup_{k \in \mathcal{K}(n-1)} B_k^{n-1}\right| = N$ it follows that the cost for computing $M_{s,p}^{n-1,k}$, $\forall k \in \mathcal{K}(n-1)$, is at most $\mathcal{O}(N)$. Furthermore, we have from Remark 9 that $|\mathbf{B}_k^{n-2}| \leqslant 8M(p)$, $\forall k \in \mathcal{K}(n-2)$.

Since for each level i, $\left|\cup_{k\in\mathcal{K}(i)} B_k^i\right| = N$, then the total cost of computing $M_{s,p}^{i,k}$, $\forall k \in \mathcal{K}(i)$, is at most $\mathcal{O}(N)$ and $|\mathbf{B}_k^{i-1}| \leqslant 8M(p)$, $\forall k \in \mathcal{K}(i-1)$. The result follows. ∎

2.6. Properties. The adapted HB construction has some interesting properties. In particular, the space $\mathbb{R}^N$ can be decomposed in a series of nested subspaces that are orthogonal to $\mathcal{P}^p(X)$ and the basis forms an orthonormal set. As a side benefit, this series of nested subspaces can be used to prove the uniqueness of the RBF interpolation problem. One important property of the adapted HB is presented in the following lemma.

LEMMA 2.11. *The basis of R^N described by the vectors of $\{\bar{C}_0^{-1}, \bar{D}_0^{-1}, \bar{D}_0^0, \bar{D}_1^0, \ldots, \bar{D}_k^n\}$, $j = 0 \ldots n$, $k \in \mathcal{K}(j)$ form an orthonormal set.*

Proof. We prove this by a simple induction argument. Assume that for level i the set of vectors $\{\mathbf{B}_k^i\}$ are orthonormal. Since the rows of the set $V_{s,p}^{i,k}$ are orthonormal and $\mathbf{B}_l^i \cap \mathbf{B}_k^i = \emptyset$ whenever $l \neq k$, then it follows that the vectors $\cup_{k\in\mathcal{K}(i)}\{\bar{C}_k^i, \bar{D}_k^i\}$ form an orthonormal basis. The result then follows from Lemma 2.6. ∎

DEFINITION 2.12. *Given a set of unisolvent interpolating nodes $X \subset \mathbb{R}^3$ with respect to $\mathcal{P}^p(\mathbb{R}^3)$, we form the matrix P from the basis vectors $\{\bar{C}_0^{-1}, \bar{D}_0^{-1}, \bar{D}_0^0, \bar{D}_1^0, \ldots, \bar{D}_k^n\}$.*

From Lemmas 2.8 and 2.11 the matrix P has the following properties

1. If $v \in \mathcal{P}^p(X)$ then $P^H v$ has $dim(\bar{C}_0^0)$ non-zero entries.
2. $PP^H = P^H P = I$.

One of the immediate implications of the existence of the HB with $p \geqslant m$ vanishing moments is that it can be used to show the uniqueness of the RBF interpolant.

THEOREM 2.13. *Suppose that the set of interpolating nodes X is unisolvent with respect to $\mathcal{P}^p(\mathbb{R}^3)$ and the kernel matrix $K(\cdot,\cdot)$ satisfies Definition 1.2, then the RBF Problem (1.2) has a unique solution.*

Proof. Since we have shown the existence of a matrix P that annihilates any polynomial in $\mathcal{P}^p(X)$, then the result follows from the argument given in section 1.1. ∎

3. Multi-Resolution RBF Representation. The HB we constructed above is adapted to the kernel and the location of the interpolation nodes. It also satisfies the vanishing moment property. The construction of such an HB leads to several important consequences. First, we can use the basis to prove the existence of a unique solution of the RBF problem, but more importantly, this basis can be used to solve the RBF problem efficiently.

As the reader might recall from section 1.1, the construction of the adapted HB decouples the polynomial interpolant from the RBF functions if the order of the vanishing moments p is equal to the order of the RBF polynomial interpolant m. This simple result can be extended if $p \geqslant m$.

THEOREM 3.1. *Suppose X is unisolvent with respect to u and u solves the interpolation problem 1.2 uniquely, where $u \perp \mathcal{P}^m$ and the kernel satisfies Definition 1.2. If the number of vanishing moments $p \geqslant m$ then*

$$\begin{pmatrix} C_\perp^H K C_\perp & C_\perp^H K T \\ T^H K C_\perp & T^H K T \end{pmatrix} \begin{pmatrix} s \\ w \end{pmatrix} = \begin{pmatrix} C_\perp^H d \\ T^H d \end{pmatrix}, \tag{3.1}$$

for some $s \in \mathbb{R}^{M-O}$ and $w \in R^{N-M}$, where $T := [\bar{D}_0^0, \bar{D}_1^0, \ldots, \bar{D}_k^n]$, $C_\perp = \bar{D}_0^{-1}$ and $O = dim(\mathcal{P}^m(X))$.

Proof. Since $u \in \mathcal{P}^p(X) \backslash \mathcal{P}^m(X) \oplus W^0 \oplus \dots W^n$, then $u = C_\perp s + Tw$ for some $s \in \mathbb{R}^{M-O}$ and $w \in R^{N-M}$, where $O = dim(\mathcal{P}^m(X))$. Replacing u into (1.2), pre-multiplying by $[C_\perp^H T^H]^H$ and recalling that $C_\perp, T \perp \mathcal{P}^m(X)$ the result follows. ∎

Once u is solved, c is easily obtained by solving the set of equations $L^H Q c = L^H(d - Ku)$, where $L^H Q \in R^{M(p)\times M(p)}$.

There are two ways we can solve this, since L and Q span the same space and have full column rank, then $L^H Q$ is invertible and

$$c = (L^H Q)^{-1} L^H (d - Ku).$$

Alternatively, we can define the interpolation problem in terms of the basis vectors in L directly i.e. $Q := L$, which leads to

$$c = L^H (d - Ku).$$

For the rest of this section we describe the algorithms for solving the previous system of equations. The entries of the matrix

$$K_W := \begin{pmatrix} C_\perp^H K C_\perp & C_\perp^H K T \\ T^H K C_\perp & T^H K T \end{pmatrix}$$

are formed from all the pairwise matching of any two vectors $\psi^i_{k,m}, \psi^j_{l,g}$ from the set $\mathcal{D} := \{\bar{D}_0^{-1}, \bar{D}_0^0, \bar{D}_1^0, \dots, \bar{D}_k^n\}$. The entries of K_W take the form

$$\sum_{k\in\mathcal{K}(n)} \sum_{k'\in\mathcal{K}(n)} \sum_{e_a\in\mathbf{B}_k^n} \sum_{e_b\in\mathbf{B}_{k'}^n} K(F_p(e_a), F_p(e_b)) \psi^i_{k,m}[F_q(e_a)] \psi^j_{l,g}[F_q(e_b)], \tag{3.2}$$

Notice that the summation is over all the vectors e_o s.t. $o = 1, \dots, N$. However, the entries of $\psi^i_{k,m}$ are mostly zeros, thus in practice the summation is over all the non-zero terms.

Continuing with the same notation, the entries of $d_W := Td$ have the form

$$\sum_{k\in\mathcal{K}(n)} \sum_{e_a\in\mathbf{B}_k^n} \psi^i_{k,m}[F_q(e_a)] f(F_p(e_a)).$$

Since $w = Pu$ and $u \perp \mathcal{P}^p(X)$, then entries of w have the form

$$\sum_{k\in\mathcal{K}(n)} \sum_{e_a\in\mathbf{B}_k^n} \psi^i_{k,m}[F_q(e_a)] u[F_q(e_a)], \;\; \forall \psi^i_{k,m} \in \mathcal{D}.$$

It is clear that from the set $\mathcal{D}$ the matrix K_W is ordered such that the entries of any row of K_W sums over the same vector $\psi^i_k \in \mathcal{D}$. In Figure 3 a block decomposition of the matrix K_W is shown.

One interesting observation of the matrix K_W is that most of the information of the matrix is contained in a few entries. Indeed, for integral equations it can be shown that an adapted HB discretization matrix requires only $\mathcal{O}(Nlog(N)^{3.5})$ entries to achieve optimal asymptotic convergence [18]. This has been the approach that was followed behind the idea of wavelet sparsification of integral equations [9, 1, 2, 18, 3, 43, 44].

However, it is not necessary to compute the entries of K_W for efficiently inverting the matrix, but instead we *only* have to compute matrix vector products of the submatrices $K_W^{i,j}$ in $\mathcal{O}(N)$ or $\mathcal{O}(Nlog(N))$ computational steps.

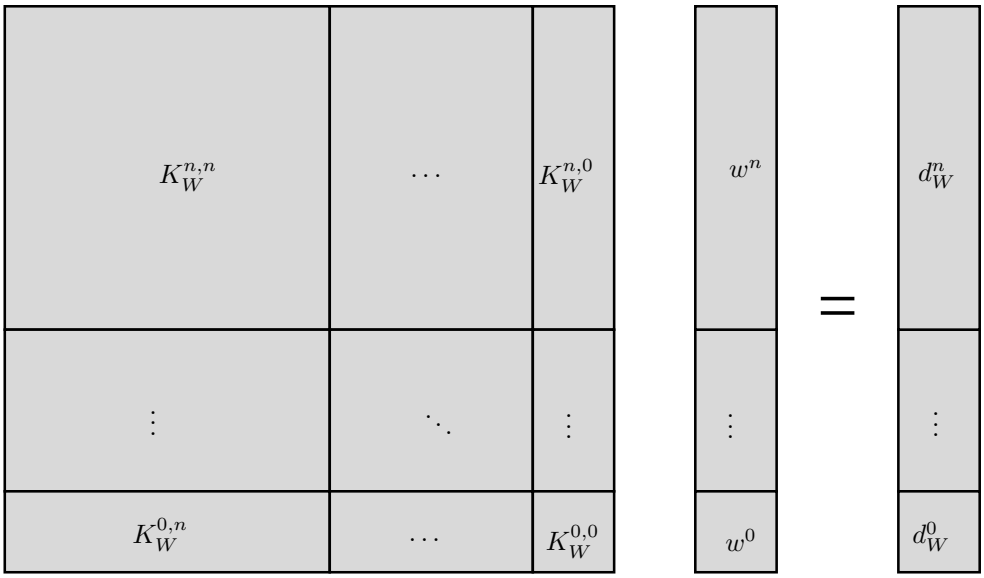


FIG. 3.1. *Organization of the linear system $K_W w = d$. The block matrices K_W^{ij} consist of all the summations in Equation 3.2, for all $\psi_{k,m}^i, \psi_{l,g}^j \in \mathcal{D}$ that belong to level i and j. The vectors d_W^i correspond to all inner products of $\psi_{k,m}^i \in \mathcal{D}$ at level i. Similarly for w, where $w = Tu$.*

3.1. Pre-conditioner. One key observation of the matrix K_W is that each of the blocks $K_W^{i,j}$ is very well conditioned. Our experiments indicate that this is the case even for highly non uniform placement of the nodes. We propose to use two kinds of preconditioners on the decoupled RBF problem: a block SSOR and a diagonal preconditioner based on the multi-resolution matrix K_W. The block SSOR multi-resolution preconditioner shows better iteration counts and is a novel approach to preconditioning. However, in practice, the simplicity of the diagonal preconditioner makes it easier to code and is faster per iteration count for the size of problems in which we are interested.

The block SSOR preconditioner on the decoupled RBF takes the form of the following problem:

$$\bar{P}^{-1} K_W w = \bar{P}^{-1} d_W, \tag{3.3}$$

where $K_W \to L_W + D_W + L_W^H$ and

$$L_W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ K_W^{1,0} & 0 & 0 & 0 \\ \vdots & \ddots & 0 & 0 \\ K_W^{n,0} & \dots & K_W^{n,n-1} & 0 \end{bmatrix} \quad \textit{and} \quad D_W = \begin{bmatrix} K_W^{0,0} & 0 & 0 & 0 \\ 0 & K_W^{1,1} & 0 & 0 \\ & & \ddots & \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & K_W^{n,n} \end{bmatrix}.$$

The block preconditioner is constructed as $\bar{P} = (L_W + D_W) D_W^{-1} (L_W^H + D_W)$.

We can solve this system of equations with a restarted GMRES (or MINRES since the matrices are symmetric) iteration [48]. To compute each iteration efficiently we need each of the matrix vector products of the blocks $K_W^{i,j}$ to be computed with

a fast summation method. We have the choice of either computing each block as matrix-vector products from a fast summation directly, or a sparse preconditioner that can be built and stored.

3.1.1. Fast Summation. It is not necessary to compute the matrix K_W directly, but to employ approximation methods to compute matrix-vector products $K_W\alpha_W$ efficiently. To such end we make the following assumption.

ASSUMPTION 2. *Let $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_{N_1} \in \mathbb{R}^3$, $c_1, c_2, \ldots c_{N_1} \in \mathbb{R}$, $R_{BF} := span(K(x, \vec{y}_1), K(x, \vec{y}_2) \ldots, K(x, \vec{y}_{N_1}))$, and $T = span\{\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_q\}$, for some set of linearly independent functions $\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_q$. We are interested in the evaluation of the RBF map*

$$\phi(x) := \sum_{i=1}^{N_1} c_i K(\vec{x}, \vec{y}_i),$$

where $\vec{x} \in \mathbb{R}^3$. Suppose there exists a transformation $F(\phi(\vec{y})) : R_{BF} \to T$ with $\mathcal{O}(N_1)$ computational and storage cost. Moreover, any successive evaluation of $F(\phi(\vec{y}))$ can be performed in $\mathcal{O}(1)$ operations and

$$|F(\phi(\cdot)) - \phi(\cdot)| \leqslant C_F A \left(\frac{1}{a}\right)^{\tilde{p}+1},$$

where $\tilde{p} \in \mathbb{Z}^+$ is the order of the fast summation method, $A = \sum_{i=1}^{N_1} |c_i|$, $C_F > 0$ and $a > 1$.

There exist several methods that satisfy, or nearly satisfy, Assumption 2. In particular we refer to those based on multi-pole expansions and the Non-equidistant Fast Fourier Transform [6, 45, 59].

The system of equations (3.3) can now be solved using an inner and outer iteration procedure. For the outer loop a GMRES algorithm is used, where the search vectors are based on the matrix $\bar{P}^{-1}K_W$.

The inner loop consists of computing efficiently the matrix-vector products $\bar{P}^{-1}K_W\alpha_W$, for some vector $\alpha_W \in \mathbb{R}^N$. This computation is broken down into two steps:

Step One To compute efficiently $K_W\alpha_W$ for each matrix vector product $K_W^{i,j}\alpha_W^j$, we fix $\psi_{k,m}^i$ from Equation (3.2) and then transform the map

$$\sum_{\psi_{l,g}^j \in \bar{D}^j} \sum_{\vec{y}_b \in X} K(\vec{x}_a, \vec{y}_b)\psi_{l,g}^j[F_q(F_p^{-1}(\vec{y}_b))]\alpha_{l,g}^j,$$

for all the vectors $\psi_{l,g}^j \in \bar{D^j}$, into a new basis $\{\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_q\}$. The computational cost for this procedure is $\mathcal{O}(N_1)$, where N_1 corresponds to the number of non-zero entries of all $\psi_{l,g}^j \in \bar{D}^j$. Since the computational cost for evaluating the new basis on any point $\vec{x}_a \in X$ is $\mathcal{O}(1)$, then the total cost for calculating each row of $\bar{K}_W^{i,j}\alpha_W^j$ is $\mathcal{O}(N_1 + N_2)$, where N_2 is equal to all the non-zero entries of $\psi_{k,m}^i$.

Now, since for each $j = 0, \ldots, n$, $\left|\cup_{k\in\mathcal{K}(j)} B_k^j\right| = N$, then N_1 is bounded by CN for some $C > 0$. For the same reason N_2 is also bounded by CN. This implies that the total cost for evaluating the matrix vector products $K_W\alpha_W$ is $\mathcal{O}\left((n+1)^2 N\right)$.

Step Two: The computation of $\bar{P}^{-1}\beta_W$, where $\beta_W := K_W\alpha_W$ is broken up into three stages. First, let $\gamma_W := (L_W^H + D_W)^{-1}\beta_W$, then

$$\begin{bmatrix} K_W^{1,1} & K_W^{1,2} & \dots & K_W^{1,n} \\ 0 & K_W^{2,2} & \ddots & \vdots \\ 0 & 0 & \ddots & K_W^{n-1,n} \\ 0 & 0 & 0 & K_W^{n,n} \end{bmatrix} \begin{bmatrix} \gamma_W^1 \\ \gamma_W^2 \\ \vdots \\ \gamma_W^1 \end{bmatrix} = \begin{bmatrix} \beta_W^1 \\ \beta_W^2 \\ \vdots \\ \beta_W^n \end{bmatrix}.$$

Since $(L_W^H + D_W)$ has a block triangular from, we can solve the inverse-matrix vector product with a back substitution scheme. Suppose that we have solved $\gamma_W^1, \dots, \gamma_W^{i-1}$, then it is easy to see from the triangular structure of $(L_W^H + D_W)$ that

$$\gamma_W^i = (K_W^{i,i})^{-1}[\alpha_W^i - \sum_{k=1}^{i-1} K_W^{i,i-1}\gamma_W^k].$$

The cost for evaluating this matrix vector product with a fast summation method is $\mathcal{O}((n+1)^2N + k(n+1)N)$. The last term comes from the block matrices in D_W, which are inverted indirectly with k Conjugate Gradient (CG) iterations [29, 25]. In Section 4 we show numerical evidence that k converges rapidly for large numbers of interpolating nodes.

The second matrix vector product, $\eta := D_W\gamma_W$ is evaluated in $\mathcal{O}(N)$ using a fast summation method. Finally the last matrix vector product $\mu := (L_W + D_W)^{-1}\eta_W$ can be solved in $\mathcal{O}((n+1)^2N + knN)$ by again using a back substitution scheme.

REMARK 10. *For many practical distributions of the interpolating nodes in the set X, the number of refinement levels $n+1$ is bounded by $C_1 \log N$ [6]. For these types of distributions the total cost for evaluating $P_W^{-1}K_W\alpha_W$ is $\mathcal{O}(Nlog^2N)$ assuming k is bounded.*

This approach is best for large scale problems where memory becomes an issue and for large vanishing moments. For small to medium size problems the blocks $K_W^{i,i}$ can be computed in sparse form and then stored for repeated use.

3.1.2. Sparse Pre-conditioners. In this section we show how to produce two types of sparse preconditioners by leveraging the ability of HB to produce compact representations of the discrete operator matrices.

The key idea is to produce a sparse matrix $\tilde{P}$ of $\bar{P}$ from the entries of the blocks $K_W^{i,j}$. This is done by choosing an appropriate strategy that decides which entries to keep, and which ones not to compute.

Although it is possible to construct an accurate approximation of $\bar{P}$ and K_W for all the blocks $K_W^{i,j}$ $(i, j = 0 \dots n)$, the computational bottleneck lies in computing the matrix vector products with $(K_W^{i,i})^{-1}$. Thus it is sufficient to compute the sparse diagonal blocks of $\tilde{K}_W$. The off-diagonal blocks are computed using the fast summation method described in section 3.1.1.

DEFINITION 3.2. *For every vector $\psi_{k,m}^i \in \mathcal{D}$ and the associated support box $B_i^m \in \mathcal{B}$, define the set L_m^i to be the union of B_i^m and all boxes in $\mathcal{B}_j$ that share a face, edge or corner with B_i^m i.e. the set of all adjacent boxes.*

To produce the sparse matrix $\tilde{P}$ we execute the following strategy: For each entry in $K_W^{i,i}$ corresponding to the adapted HB vectors $\psi_{k,m}^i$, $\psi_{l,g}^i \in \mathcal{D}$, we only compute this entry if

$$dist(L_k^i, L_l^i) := \inf_{\vec{x},\vec{y}} \|L_k^i(\vec{x}) - L_l^i(\vec{y})\|_{L^2(\mathbb{R}^3)} \leqslant \tau_{i,i}, \tag{3.4}$$

where $\tau_{i,i} \in R^+$ for $i = 0 \dots n$. For an appropriate distance criterion $\tau_{i,i}$ we can produce a highly sparse matrix $\tilde{K}_W^{i,i}$ that is close to $K_W^{i,j}$ (and respectively $\tilde{P}$) in a matrix 2-norm sense.

DEFINITION 3.3. *The distance criterion $\tau_{i,j}$ is set to*

$$\tau_{i,j} := 2^{n-i}, \tag{3.5}$$

With this distance criterion it is now possible to compute a sparse representation of the diagonal blocks of K_W. This can be seen from the following lemma.

LEMMA 3.4. *Let B_a be a ball in $\mathbb{R}^3$ with radii r_a centered around the midpoint $a \in \mathbb{R}^3$ of the box B_i^m. Similarly, let B_b be a ball in $\mathbb{R}^3$ with radii r_b centered around the midpoint b of the box B_l^j. Suppose $\psi_{k,m}^i$ and $\psi_{l,g}^j$ satisfy the vanishing moment condition Equation 2.1 for all $r \in \mathcal{P}^p(X)$, and $B_m^i \subset B_a$, $B_l^j \subset B_b$. If $K(\vec{x}, \vec{y})$ satisfies Assumption 1, then the following estimate holds:*

$$|a(\psi_{k,m}^i, \psi_{l,g}^j)| \leqslant \frac{NC}{dist(B_a, B_b)^{q+(p+1)}} \frac{(p+1) r_a^{p+1}}{p!}.$$

where $a(\psi_{k,m}^i, \psi_{l,g}^j)$ is equal to

$$\sum_{k \in \mathcal{K}(n)} \sum_{k' \in \mathcal{K}(n)} \sum_{e_a \in \mathbf{B}_k^n} \sum_{e_b \in \mathbf{B}_{k'}^n} K(F_p(e_a), F_p(e_b)) \psi_{k,m}^i [F_q(e_a)] \psi_{l,g}^j [F_q(e_b)]. \tag{3.6}$$

Proof. Since we assume that $K(\vec{x}, \vec{y})$ is analytical, then by Taylor's theorem we have that for every $\vec{x} \in B_a$

$$K(\vec{x}, \vec{y}) = \sum_{|\alpha|=0}^{p} \frac{D_{\vec{x}}^\alpha K(a, \vec{y})}{\alpha!} (\vec{x} - a)^\alpha + R_\alpha(\vec{x}, \vec{y}),$$

where $(\vec{x} - a)^\alpha := (x_1 - a_1)^{\alpha_1} (x_2 - a_2)^{\alpha_2} (x_3 - a_3)^{\alpha_3}$, $\alpha! := \alpha_1! \alpha_2! \alpha_3!$,

$$R_\alpha(\vec{x}, \vec{y}) := \sum_{|\alpha|=p+1} \frac{(\vec{x} - a)^\alpha}{\alpha!} \int_0^1 D_{\vec{x}}^\alpha K_f(a + s(\vec{x} - a), \vec{y})\, ds \tag{3.7}$$

and $s \in [0, 1]$. Now recall the vanishing moment property of the basis, then Equation (3.6) becomes

$$\sum_{k \in \mathcal{K}(n)} \sum_{k' \in \mathcal{K}(n)} \sum_{e_a \in \mathbf{B}_k^n} \sum_{e_b \in \mathbf{B}_{k'}^n} R_\alpha(F_p(e_a), F_p(e_b)) \psi_{k,m}^i [F_q(\vec{x}_a)] \psi_{l,g}^j [F_q(\vec{y}_b)].$$

Now, let $z = a + s(\vec{x}_a - a)$. By applying the chain rule and Assumption 1 we obtain

$$|D_{\vec{x}}^\alpha K(\vec{x}, \vec{y})| \leqslant \frac{C s^\alpha}{|z - w|^{q+|\alpha|}}.$$

Since $z \in B_a$ and $\vec{y} \in B_b$ then

$$|D_{\vec{x}}^{\alpha} K(\vec{x}, \vec{y})| \leqslant \frac{C s^{\alpha}}{dist(B_a, B_b)^{q+|\alpha|}}, \tag{3.8}$$

where $dist(B_a, B_b) := \inf_{\vec{x} \in B_a, \vec{y} \in B_b} |\vec{x} - \vec{y}|$, i.e. the minimum distance between the two balls B_a and B_b. Replacing equation (3.8) into (3.7), then $|R_\alpha(\vec{x}_a, \vec{y}_b)|$ is bounded by

$$\begin{aligned} |R_\alpha(\vec{x}_a, \vec{y}_b)| &\leqslant \sum_{|\alpha|=p+1} \frac{C}{dist(B_a, B_b)^{q+|\alpha|}} \left| \frac{(\vec{x} - a)^\alpha}{\alpha!} \right| \int_0^1 s^\alpha \, ds \\ &\leqslant \frac{C}{dist(B_a, B_b)^{q+(p+1)}} \frac{(p+1) r_a^{p+1}}{p!}. \end{aligned} \tag{3.9}$$

From the orthonormality of $\psi_{k,m}^i$ and $\psi_{l,g}^j$ the result follows.
□

The previous lemma shows the decay of the entries of the matrix $\tilde{K}_W$ is dependent on the distance between the respective blocks and the number of vanishing moments. If p is chosen sufficiently large (for a biharmonic $p = 3$ is sufficient), the entries of $\tilde{K}_W$ decay polynomially fast, which leads to a good approximation to K_W.

Under this sparsification strategy, it can be shown that $\|K_W - \tilde{K}_W\|_2$ decays exponentially fast as a function of the number of vanishing moments p with only $\mathcal{O}(N(n+1)^2)$ entries in $\tilde{K}_W$. The accuracy results have been derived in more detail in an upcoming paper we are writing for anisotropic spatially varying RBF interpolation [19].

LEMMA 3.5. *Let $\mathbb{N}(A) : \mathbb{R}^{N \times N} \to \mathbb{R}^+$, be the number of non-zero entries for the matrix A, then we have*

$$\mathbb{N}(\tilde{K}_W^{i,i}) \leqslant 8M(p)7^3 N \tag{3.10}$$

Proof. First, identify the box $L_{k,m}^i$ that embeds $\psi_{k,m}^i$ and the distance criterion $\tau_{i,i}$ associated with that box. Now, the number of vectors $\psi_{l,g}^i$ and corresponding embedding $L_{l,g}^i$ that intersect the boundary traced by τ is equal to $(2^{-i}3 + 2\tau_{i,i} + 2^{-i+1})^3 / 2^{-i} \leqslant 2^{3(i-i)} 7^3 = 7^3$ (as shown in Figure 3.2). From Remark 9 there are at most $8M(p)$ HB vectors per cube. The result follows. □

To compute the block diagonal entries of $\tilde{K}_W^{i,i}$, for $i = 0, \ldots, n$ in $\mathcal{O}(N \log N)$ computational steps, we employ a strategy similar to the fast summation strategy in section 3.1.1. For each row of $\tilde{K}_W^{i,i}$, locate the corresponding HB $\psi_{k,m}^i$ from Equation (3.2) and transform the map

$$\sum_{k \in \mathcal{K}(n)} \sum_{k' \in \mathcal{K}(n)} \sum_{e_a \in \mathbf{B}_k^n} K(F_p(\vec{x}, e_a)) \psi_{k,m}^i [F_q(e_a)] \tag{3.11}$$

into an approximation $G(\vec{x}, \psi_{k,m}^i) := \sum_{i=1}^q c_i^{\psi_{k,m}^i} \tilde{\phi}_i$ by applying a fast summation method that satisfies Assumption 2. Any entry of the form $a(\psi_{k,m}^i, \psi_{l,g}^i)$ can be

computed by sampling $G(\vec{x})$ at locations corresponding to the non-zero entries of $\psi^i_{l,g}$, and the sampled values can be used to multiply and sum through the non-zero values of $\psi^i_{l,g}$.

THEOREM 3.6. *Each block $\tilde{K}^{i,i}_W$ is computed in $\mathcal{O}(N)$ steps.*

Proof. The cost for computing the basis of $G(\vec{x})$ corresponding to $\psi^i_{k,m}$ is at most $\mathcal{O}(N^i_{k,m})$, where $N^i_{k,m}$ is the number of non zeros of $\psi^i_{k,m}$. Now, since $\left|\cup_{k\in\mathcal{K}(i)} B^i_k\right| = N$ the cost of computing $G(\vec{x}, \psi^i_{k,m})$ for all the vectors $\psi^i_{k,m}$ at level i is $\sum_{\psi^i_{k,m}\in D^i_k, k\in\mathcal{K}(i)} N^i_{k,m} = \mathcal{O}(N)$.

For each row in $\tilde{K}^{i,i}_W$, from Lemma 3.5 there is at most $8M(p)7^3$ entries. This implies that for each vector $\psi^i_{k,m}$ we need only $\mathcal{O}(1)$ evaluations of $G(\vec{x}, \psi^i_{k,m})$ to compute a row of $\tilde{K}^{i,i}_W$. Now, if we sum up the cost of evaluating $G(\vec{x}, \psi^i_{k,m})$ for all the rows then the total cost for evaluating $\tilde{K}^{i,i}_W$ is $\mathcal{O}(N)$. ∎

REMARK 11. *For each entry in $\tilde{K}^{i,i}_W$, the corresponding basis vectors $\psi^i_{k,m}, \psi^j_{l,g}$ can be found in $\mathcal{O}(n)$ computational steps. This is easily achieved by sorting the set of cubes $\{B^j_l\}_{l\in\mathcal{K}, j=1,\dots,n}$ with an octree structure, i.e. a parent-child sorting.*

REMARK 12. *Note that further improvements in computation can be done by observing that $\psi^i_{k,m}$ is a linear combination of the vectors $\phi^{i-1}_{l,o} \in V^{i-1}$. Thus equation (3.11) can be written as a linear combination of*

$$\sum_{k\in\mathcal{K}(n)} \sum_{k'\in\mathcal{K}(n)} \sum_{e_a\in\mathbf{B}^n_k} K(\vec{x}, F_p(e_a))\phi^{i-1}_{l,o}[F_q(e_a)]. \tag{3.12}$$

If two vectors $\psi^i_{k,m}$ and $\psi^i_{k,m'}$ are in the same cube then it is sufficient to compute equation (3.11) once and apply the coefficients computed in the construction of the entire HB. In addition, if two vectors $\psi^i_{k,m}$ and $\psi^i_{k',m'}$ share the same vector $\phi^i_{l,o} \in V^{i-1}$, the same procedure can be applied. In our results in Section 4 we apply this scheme to compute the SSOR and diagonal blocks.

REMARK 13. *As our results show a very simple diagonal preconditioner can be built from the blocks of $K^{i,i}_W$. In particular*

$$P := diag\left(\begin{bmatrix} K^{1,1}_W & 0 & \dots & 0 \\ 0 & K^{2,2}_W & \ddots & \vdots \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & K^{n,n}_W \end{bmatrix}\right).$$

This preconditioner is also much easier to construct in practice.

4. Numerical Results. In this section we apply the multi-resolution method developed in section 3 to the RBF interpolation and regression problem. These will be of different sizes and polynomial orders for the biharmonic, multiquadric and inverse multiquadric function kernel in $\mathbb{R}^3$. These kernels can be written in a common form $K(r) := (r^2 + \delta^2)^{l/2}$, where $r := |x|$, $\delta \in \mathbb{R}$ and $l \in \mathbb{Z}$. The RBF solver is also tested on a generalized least squares and Kriging problem. The distribution of the nodes in X are separated into three cases.

Test Case 1: We test our method on several sets of randomly generated interpolating nodes in the unit cube in $\mathbb{R}^3$ as shown in Figure 3.4(a). The sets of interpolating nodes $\{X_1, \dots, X_r\}$ vary from 1000 to 400,000 nodes. Each set of interpolating nodes

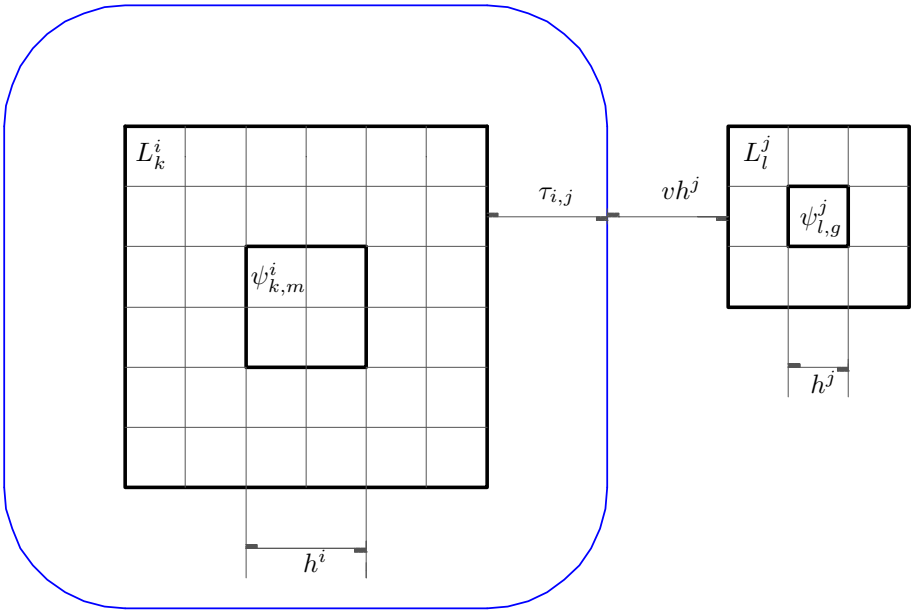


FIG. 3.2. *Distance criterion cut-off boundary for the cube L_k^i, corresponding to all the vectors $\psi_{k,m}^i \in \bar{D}_k^i$. Assume $j \geqslant i$ and $h = 2^{-1}$, and each cube B_k^i is evenly divided by B_l^j. With this in mind, the cut-off criterion traces a cube of length $2\tau_{i,j}$ plus the side length of L_l^j. For any vector $\psi_{l,g}^j$ such that L_l^j crosses the cut-off boundary, we compute the corresponding entries in the matrix $\tilde{K}_W^{i,i}$. As v increases, by the error estimates (3.9), the entries $a(\psi_{k,m}^i, \psi_{l,g}^j)$ decay as a polynomial of p with respect to the distance vh^j.*

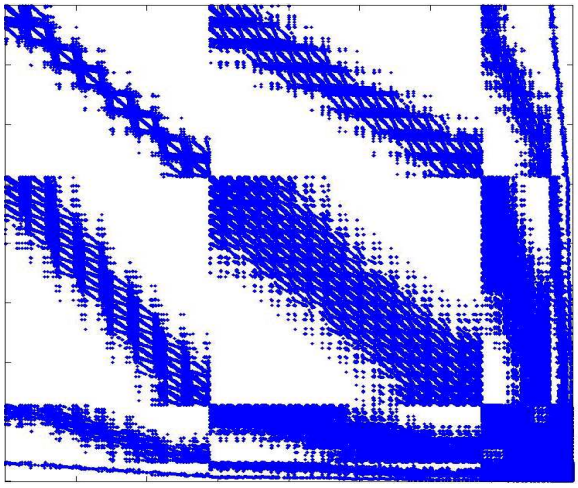
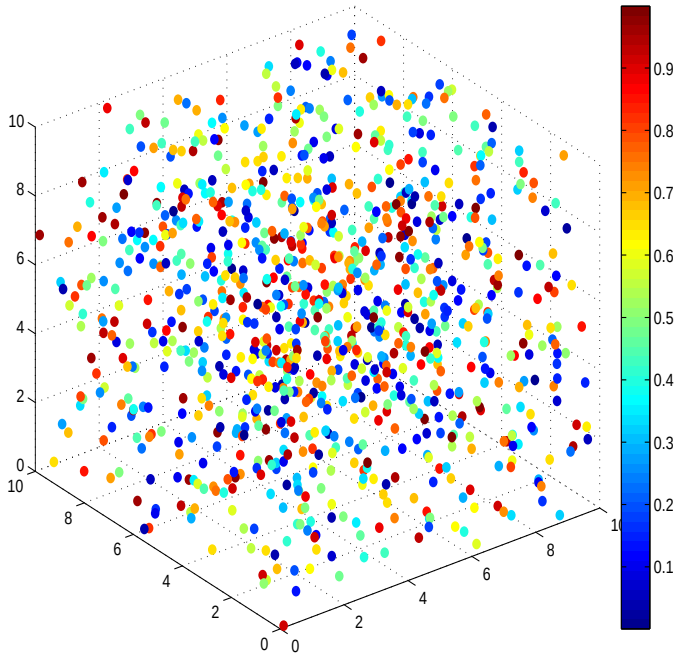
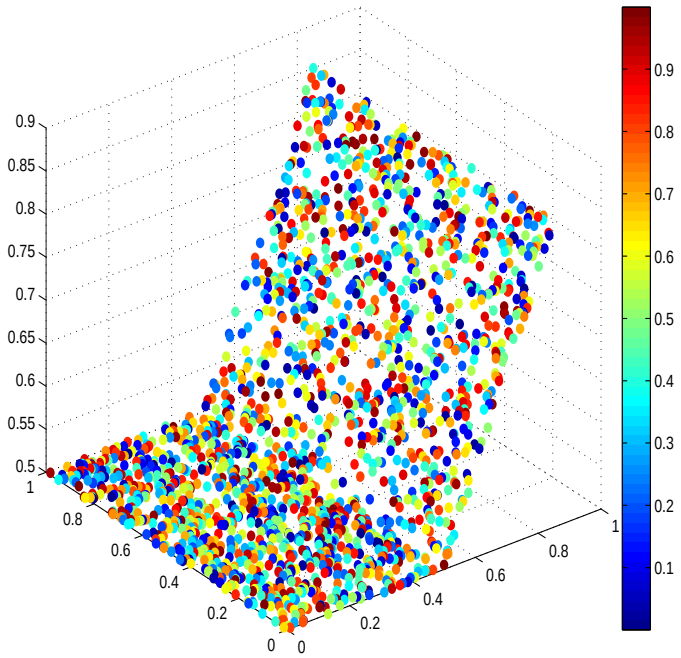


FIG. 3.3. *Example of sparsity pattern for $\tilde{K}_W$ created by applying the distance criterion to each of the blocks of $K_W^{i,j}$.*



(a)



(b)

FIG. 3.4. *(a)* **RBF interpolating set:** *Interpolating set with a thousand nodes for Test Case 1. The function value defined on each node is mapped from the color bar. (b)* **RBF interpolating set:** *V-plane example interpolating set with four thousand nodes. The interpolating nodes are obtained by projecting the data nodes from Test Case 1 onto two intersecting planes. The function values on each node is randomly assigned.*

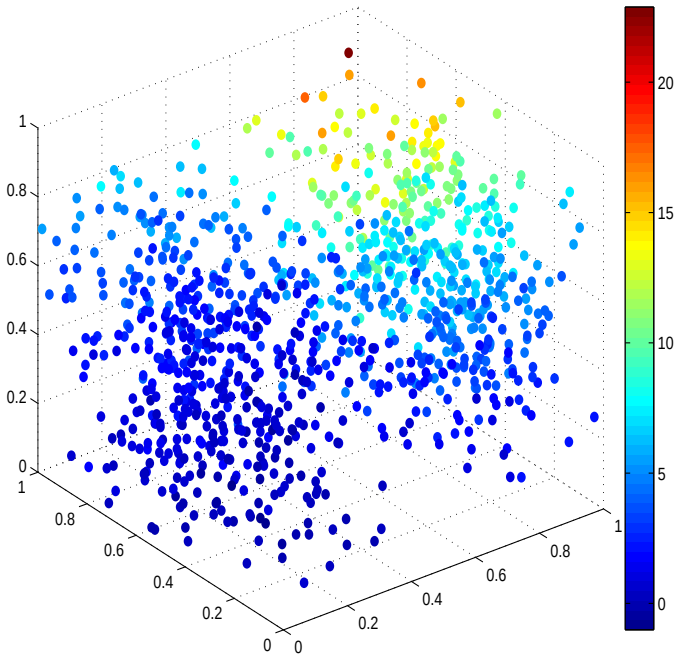


FIG. 3.5. *Test Case 3. Interpolating set of 1000 centers. The nodes locations are placed with a bimodal distribution. The color scheme indicates the value of the observations.*

is a subset of any other set with bigger cardinality, i.e., $X_l \subset X_{l+1}$. The function values on each node are also grouped into r sets $\{b_1, \ldots, b_r\}$ with randomly chosen values and satisfy also $b_l \subset b_{l+1}$.

Test Case 2: For this second test we apply a projection of the data nodes generated in *Test Case 1* onto two non-orthogonal planes $\mathbb{R}^3$, then remove any *two* nodes that are less than 10^{-4} distance from each other. The V-plane intersecting are shown in Figure 3.4(b). Note, that only about 0.1% of the centers were eliminated and the number of nodes in the table is approximate.

Test Case 3: For the third test, we still use interpolating nodes in the unit cube in 3D. But instead of having the nodes uniformly distributed as in *Test Case 1* and *2*, we use a bimodal Gaussian distribution with a standard deviation of 0.25 and mean vectors of $(0, 0.5, 0.5)$ and $(1, 0.5, 0.5)$, respectively (see Figure 3.5). For the right-hand-side we assume that the observations come from a third-degree polynomial, contaminated by a random noise with the covariance matrix defined by the inverse multiquadric RBF. This is an example where the RBF problem can be recast as a special case of the generalized least squares problem, also known as Kriging [33, 36]. In this type of problems, the order of the polynomial set $\mathcal{P}^m(X)$ is associated with the dimension of independent variables in the regression model. Hence, the ability of the RBF solver to deal with higher-order polynomials become essential, if a higher-dimensional regression model for Kriging is to be considered.

Test Setup: The implementation of the multi-resolution discrete HB method is performed in C++. The GMRES algorithm is incorporated from PETSc (Portable, Extensible Toolkit for Scientific computation) libraries [4] into our C++ code. Inner and outer iterations are solved using a GMRES algorithm with 100-iteration restart.

In the rest of this section when we refer to GMRES iterations, we imply restarted GMRES with a restart for every 100 iterations. Since the preconditioned system will introduce errors in the RBF residual of the original Problem 1.2, the accuracy of the GMRES is adjusted such that the residual ϵ of the unpreconditioned RBF system is less than 10^{-3}. For the results shown in Tables 4.2, 4.3, 4.4 and 4.5, we show the GMRES accuracy residual.

All the numerical tests with a fast summation method are performed with a single processor version of the Kernel-Independent Fast Multipole Method (KIFMM) 3D code (http://mrl.nyu.edu/~harper/kifmm3d/documentation/index.html). This code implements the algorithm described in [59]. The accuracy is set to relative medium accuracy (10^{-6} to 10^{-8}).

All simulations were performed on a single core on the TACC Lonestar cluster (http://www.tacc.utexas.edu/services/userguides/lonestar/). Each node has two processors. Each is a Xeon 5100 series 2.66GHz dual-core processor with a 4MB unified (Smart) L2 cache. Peak performance for the four cores is 42.6 GFLOPS.

Test Examples:

- *Condition number κ of underlying system of equations with respect to scaling all the domain.* One immediate advantage our method has over a direct method is the invariance of the conditioning of the system of equations with respect to the scale of the polynomial domain. This is a consequence of the construction of the HB polynomial orthogonal basis.
  Removing the polynomial source of ill-conditioning makes the system easier to solve. The condition number of the full RBF interpolation matrix is sensitive to the scaling of the domain. We show this by scaling the domain with respect to the polynomial interpolation and fix the scaling for the RBF part:

$$\begin{pmatrix} K & \alpha Q \\ \alpha Q^T & O \end{pmatrix} \begin{pmatrix} u \\ c \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}.$$

  As shown in Table 4.1 the condition number for a 1000 center problem with $m = 3$ and $\phi(r) = r$ deteriorates quite rapidly with scale α. In particular, for a scaling of 1000 or larger an iterative method, such as GMRES or CG, stagnates. We note that the invariance of the condition number of the decoupled system was also observed in [8, 53] and proven in [8].
  Another important observation is that the same result will apply for a multiquadric, or inverse multiquadric of the form $\phi = (r^2 + \delta^2)^{\pm l}, \delta \in \mathbb{R}$, due to the polynomial decoupling from the RBF matrix. In general, this will be true for any strictly conditionally positive (or negative) definite RBF. However, the matrix K_W will still be subject to the underlying condition number of K. In other words, if $\kappa(K)$ deteriorates significantly then K_W will be ill-conditioned also.

Scale	$\alpha = 1$	$\alpha = 10$	$\alpha = 100$	$\alpha = 1000$	$\alpha = 10000$
κ of RBF system	9×10^5	8×10^7	8×10^{11}	8×10^{14}	9×10^{18}
$\kappa(K_W)$	762	762	762	762	762

TABLE 4.1

Condition number for RBF system matrix (equation 1.2) versus scale of the problem for a thousand nodes for Test Case 1 with respect to the biharmonic $\phi(r) := r$. As observed, increasing the scale by alpha the condition number deteriorates very rapidly. In particular, for a condition number higher than the reciprocal of machine position and the GMRES or CG algorithm stagnates.

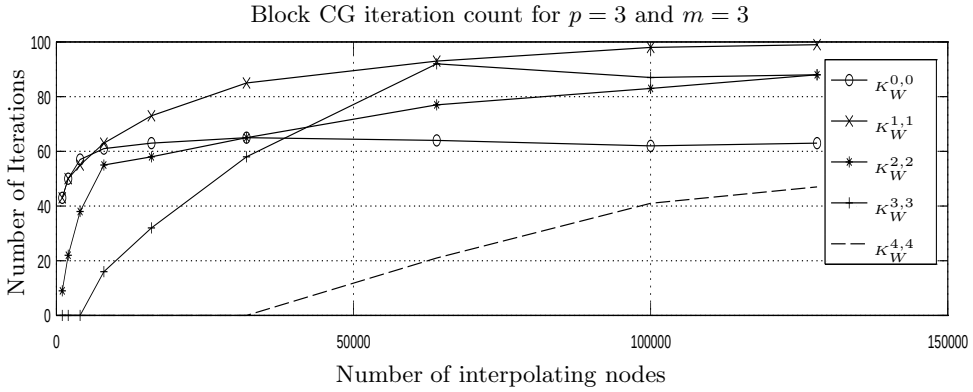


FIG. 4.1. *Inner CG block iterations for the case $p = 3$ and $m = 3$ and outer GMRES iteration counts for multiple cases. As observed the inner CG iteration count, at 10^{-6} residual relative error, for each block increases rapidly with size, but eventually stabilizes. The sparse blocks $K_W^{i,i}, i = 0, \ldots, 4$ lead to the same iteration counts.*

- *Biharmonic RBF, $m = 3$ (cubic) and $p = 3$* This is an example of a higher order polynomial RBF interpolation. We test both the SSOR and diagonal pre-conditioner on Test Case 1 & 2. For the pre-conditioner the accuracy of the GMRES outer iterations is set such that the residual $\epsilon := \|K_W w - d_W\| \leqslant 10^{-3}$. For the pre-conditioner the number of iterations to invert blocks $K_W^{i,i}$, $i = 0, \ldots, n$ with an accuracy of 10^{-6} are shown in Figure 4.1. Due to the condition number of the blocks they quickly converge with either a CG, or a GMRES solver. Moreover, the number of iterations appear to grow very slowly with size.
  From the results in Table 4.2 (a) for Test Case 1, the number of restarted GMRES iterations grows as $\mathcal{O}(N^{0.43})$. But, observe that for the changes from N = 4,000 to N = 8,000 and from N = 32,000 to N = 64,000 the complexity increases by a factor of 5. We suspect that the Level 1 and 2 cache are used less efficiently at this point due to the size of the problem. This increase is shown both in the cost per iteration and inner block computation. Fitting a linear regression function to the log-log plot leads to a growth of $N^{1.5}$ for time complexity.
  In Table 4.2 (b) the iteration and timing results for the sparse SSOR pre-conditioner for test cast 2 (v-plane) are shown. This is a much harder problem due to the corner and the projection of the random data from Test Case 1 onto two planes at 135 degrees from each other. The total GMRES iteration grows as $CN^{0.54}$, which leads to a $N^{1.54}log^2N$ time complexity.
  In Table 4.2 (c) the iteration and timing results for the diagonal preconditioner with Test Cast 1 (uniform cube) are shown. We can observe that although the GMRES iteration count is higher than that of the SSOR pre-conditioner ($CN^{0.51}$), the simplicity of the pre-conditioner allows every matrix-vector product to be computed much faster. Fitting a line to the log data leads to a total time complexity increases of $CN^{1.32}$. The memory constraints

are also much lower than the SSOR since only N entries are needed to be stored for the pre-conditioner. It requires only 5613 seconds to solve the GMRES iteration for the 100,000 center problem. The diagonal pre-conditioner is built in 7747 seconds and the total time complexity is 3.8 hours.
In Table 4.2 (d), the iteration and timing results for the diagonal pre-conditioner with test cast 2 (v-plane) are shown. This is a harder problem than Test Case 1 and can be reflected in the time complexity increase ($CN^{1.34}$) and the GMRES iteration count ($CN^{0.72}$). Another observation is that the time required to compute the diagonal pre-conditioner is about one half compared to Test Case 1. This is due to the adaptive way we compute the diagonal, recall Remark 12.

- *Biharmonic RBF, $m = 0$ (constant), $p = 3$* From Table 4.3 we can observe that the results for Test Case 1 and 2 are almost the same as for case $m = 3, p = 3$ and the GMRES iterations are almost identical. This indicates that the efficiency of our method is invariant under polynomial interpolation.
- *Biharmonic RBF $m = 1$ (linear) and $p = 3$.* For this case we only present the diagonal pre-condition since overall it is more effective than the SSOR. In Table 4.4 (a) the iterations and timing results for Test Case 1 are shown. An intriguing result is that the number of GMRES iterations are exactly the same for each case $m = 0, 1$ or 3. The iteration and total timing results are also very close and have almost the same growth. This is also observed in the iterations and timing results for diagonal-precondition for Test Case 2.
- *Multiquadric and inverse multiquadric RBF, $m = 3$ (cubic), $p = 3$.* For the case of the multiquadrics with $\delta = 0.01$, the iteration count increases significantly, as shown in Table 4.5(a). The number of GMRES iterations increases as $CN^{0.67}$. This is a harder problem to solve due to the ill-conditioning introduced by the constant term δ, as reflected by the increase in the number of GMRES iterations. Fitting a line through the log-log plot of the total time leads to a $CN^{1.21}$ time complexity. Although if we increase the number of interpolating nodes, this would probably match with the biharmonic case.
In contrast, the inverse multiquadrics result shown in Table 4.5(b) is a better conditioned problem leading to around the same complexity as for the biharmonic case, but the constant is lower. We note that to achieve comparable interpolation accuracy, the value of δ for the inverse multiquadric generally needs to be larger than for the multiquadric case. And the larger the δ the more ill-conditioned the RBF interpolation problem.
- *Regression* In this part we demonstrate how our code can be used to compute a regression problem. Let $(\Omega, \mathcal{F}, P)$ be a complete probability space. Suppose that $\varepsilon \in \mathbb{R}^N \times \Omega$ is a random vector such that $E[\varepsilon] = 0$ and $E[\varepsilon\varepsilon^H] = K$, where the covariance matrix K is assumed to be symmetric positive definite. Given a set of observations (realizations) Y, the regression problem is modeled as

$$y(\vec{x}) = \sum_{i=1}^{N} c[i] q_i(\vec{x}) + \varepsilon(\vec{x}).$$

or in matrix form

$$y = Qc + \varepsilon.$$

The generalized least squares solution is $c = (Q^H K^{-1} Q)^{-1} Q^H K^{-1} Y$ [33].

N	HB (s)	$\tilde{K}_W^{i,i}(s)$	GMRES	GMRES ($\frac{s}{iter}$)	Total (s)
1000	67	3	6	1	75
2000	68	12	8	2	93
4000	70	31	11	4	143
8000	99	259	15	21	665
16000	123	1024	20	36	1857
32000	137	3612	26	81	5855
64000	224	10388	38	387	25321
100000	326	19662	45	500	42486

(a) SSOR Pre-conditioner Test Case 1

N	HB (s)	$\tilde{K}_W^{i,i}(s)$	GMRES	GMRES ($\frac{s}{iter}$)	Total (s)
1000	69	70	13	2	92
2000	70	80	19	3	142
4000	70	101	25	17	528
8000	79	398	34	37	1589
16000	91	1146	55	102	6699
32000	110	3756	87	230	23722

(b) SSOR Pre-conditioner Test Case 2

N	GMRES	GMRES residual	Itr (s)	Diag.(s)	Total (s)
1000	33	1.22E-04	4	70	74
2000	45	5.18E-05	11	73	84
4000	64	2.73E-05	34	83	117
8000	90	1.15E-05	145	217	362
16000	126	4.98E-06	287	507	794
32000	188	1.84E-06	965	1084	2049
64000	281	1.16E-06	4080	3583	7664
100000	319	6.68E-07	5613	7747	13361
200000	491	2.46E-07	18135	21716	39851
400000	710	1.21E-07	68921	61916	130837

(c) Diag. Pre. Test Case 1

N	GMRES	GMRES residual	Itr (s)	Diag.(s)	Total (s)
1000	73	5.74E-05	7	68	75
2000	102	2.26E-05	23	71	94
4000	168	6.97E-06	69	78	146
8000	272	4.73E-06	247	156	403
16000	348	1.18E-06	569	297	866
32000	921	2.45E-07	3776	650	4425
64000	1075	1.26E-07	7225	1877	9102
100000	2209	6.39E-08	27248	3434	30683

(d) Diag. Pre. Test Case 2

TABLE 4.2

Results for biharmonic $\phi(r) := r$, $m = 3$ (Cubic), $p = 3$. (a) Iteration and timing results for the sparse SSOR pre-conditioner for test cast 1 (uniform cube). The first column is the problem size round up to the 1000's; the second the time in seconds to compute the HB. The third is to compute the sparse inner blocks $K_W^{i,i}$; the fourth is the number of iterations such that $\epsilon \leqslant 10^{-3}$ and the fifth is the time per GMRES iteration. (b) Iteration and timing results for the sparse SSOR pre-conditioner for test cast 2 (v-plane). (c) Iteration and timing results for diagonal pre-conditioner for test cast 1 (uniform cube). The first column is the problem size; the second is the number of iterations such that $\epsilon \leqslant 10^{-3}$; the third column is the GMRES residual needed to achieve $\epsilon \leqslant 10^{-3}$ accuracy; the fourth is the total time for the GMRES iteration and the fifth is the total time in seconds to compute the diagonal of all inner blocks $K_W^{i,i}$. The last column is the total time in seconds. (d) Iteration and timing results for diagonal pre-conditioner for test cast 2 (v-plane).

N	GMRES	GMRES residual	Itr (s)	Diag.(s)	Total (s)
1000	33	1.22E-04	4	68	72
2000	45	5.18E-05	11	71	82
4000	65	2.73E-05	34	83	117
8000	89	1.15E-05	130	216	345
16000	126	4.98E-06	287	505	792
32000	188	1.84E-06	968	1081	2050
64000	281	1.16E-06	4060	3588	7648
100000	319	6.68E-07	5514	7746	13261
200000	491	2.46E-07	18135	16417	35807
400000	710	1.21E-07	68976	61829	130806

(a) Diag. Pre. Test Case 1

N	GMRES	GMRES residual	Itr (s)	Diag.(s)	Total (s)
1000	77	5.78E-05	7	68	76
2000	103	2.26E-05	23	71	94
4000	174	6.97E-06	71	78	147
8000	271	4.73E-06	246	156	427
16000	374	1.18E-06	612	296	884
32000	901	2.45E-07	3313	650	3966
64000	1183	1.26E-07	7858	1876	9734
100000	2173	6.39E-08	26904	3436	30341

(b) Diag. Pre. Test Case 2

TABLE 4.3

Results for biharmonic $\phi(r) := r$, $m = 0$ (constant), $p = 3$. (a) From this test case we observe that the number of iterations for the GMRES iteration is exactly the same as for $m = 3$. There are differences in the construction of the diagonal-preconditioner and the matrix vector products with K_W. However, the total time complexity is almost the same. (b) Iteration count and timing results for pre-conditioner for Test Case 2(V-plane).

However, this is the same solution for Problem 1.2 [39] if we assume that $K(r) := \phi(r)$ and K is symmetric positive definite.
The regression can now be extended to incorporate information on the residual. First, Define a linear predictor $\hat{y}(\vec{x})$ as

$$\hat{y}(\vec{x}) = \sum_{i=1}^{N} y(\vec{x}_i)\gamma[i](\vec{x}),$$

where $\gamma \in \mathbb{R}^N$ and then minimize the mean squared error $E[(\hat{y}(\vec{x}) - y(\vec{x}))^2]$. We can then derive a Best Linear Unbiased Estimator (BLUE) $\hat{y}^*(\vec{x})$,

$$\hat{y}^*(\vec{x}) = \sum_{i=1}^{N} c[i]q_i(\vec{x}) + u[i]\phi_i(\vec{x}),$$

where c_i and u_i are solutions to the RBF interpolation problem (1.2) with $d_i = y(\vec{x}_i)$ and $K(r) = \phi(r)$ [33]. Of course, this regression model will only be valid for RBF kernels that satisfy the positive definiteness of the covariance matrix R. We note that many decaying RBFs, such as Gaussian and inverse multiquadric, satisfy this condition [37].

N	GMRES	GMRES residual	Itr (s)	Diag.(s)	Total (s)
1000	33	1.22E-04	4	68	74
2000	45	5.18E-05	11	71	84
4000	65	2.73E-05	34	78	116
8000	89	1.15E-05	130	156	343
16000	126	4.98E-06	288	297	793
32000	188	1.84E-06	951	651	2043
64000	281	1.16E-06	4062	358	7642
100000	319	6.68E-07	6011	7746	13750
200000	491	2.46E-07	16412	19378	35792
400000	710	1.21E-07	68962	61769	130731

(a) Diag. Pre. Test Case 1

N	GMRES	GMRES residual	Itr (s)	Diag.(s)	Total (s)
1000	77	5.74E-05	7	68	76
2000	103	2.26E-05	23	71	94
4000	170	6.97E-06	69	78	147
8000	298	4.73E-06	271	156	427
16000	357	1.18E-06	588	297	884
32000	901	2.45E-07	3315	651	3966
64000	1183	1.26E-07	7856	1877	9734
100000	2173	6.39E-08	26851	3433	30285

(b) Diag. Pre. Test Case 2

TABLE 4.4
Results for biharmonic $\phi(r) := r$, $m = 1$ (linear), $p = 3$. (b) Iteration count and timing results for pre-conditioner for Test Case 2(V-plane).

Here for illustration purposes, we assume that the covariance matrix is defined by the inverse multiquadric RBF, $K(r) = \delta(r^2 + \delta^2)^{-\frac{1}{2}}$. The observations are simulated as

$$Y(\vec{x}_i) = \|\vec{x}_i\|_1^3 + \varepsilon(\vec{x}_i). \tag{4.1}$$

In Table 4.6, we show the timing result of solving the Kriging regression problem for *Test Case 3* with $\delta = 0.01$ using our proposed method for diagonal pre-conditioner. The result shows linear growth of total time with respect to the problem size. Although for a larger coefficient δ the system can become ill-conditioned and our solver would not be as efficient. Note that due to the difficulty of generating large random vectors with RBF covariance matrix, we limited the size to 16,000 centers.

In the BLUE estimation problem, in general it is generally assumed that the canonical representation of the covariance matrix is known, but the scale δ is unknown. The BLUE regression problem now becomes solving for $\hat{y}$ and the coefficient δ such that

$$\delta^* = argmin_\delta E[(\hat{y}(\vec{x}, \delta) - y(\vec{x})^2].$$

We test the BLUE estimating problem assuming that the observation data is generated by the noise model in Equation (4.1) with $\delta = 0.01$. The regression

N	GMRES	GMRES residual	Itr (s)	Total (s)
1000	40	1.67E-04	2	72
2000	57	3.94E-05	13	84
4000	92	2.31E-05	36	117
8000	130	8.94E-06	156	303
16000	197	3.40E-06	603	902
32000	341	6.53E-06	1977	2527
64000	683	9.23E-07	3899	5400
100000	852	2.96E-07	9845	13003

(a)

N	GMRES	GMRES residual	Itr (s)	Total (s)
1000	9	3.12E-04	1	70
2000	7	2.59E-03	1	73
4000	14	9.52E-04	5	90
8000	18	8.73E-04	15	180
16000	27	1.08E-03	43	402
32000	63	1.01E-03	192	912
64000	85	8.62E-04	682	2735
100000	109	4.37E-04	1440	5830

(b)

TABLE 4.5

Iteration and timing results for diagonal pre-conditioner, multiquadric $\phi(r) := (r^2 + 0.01^2)^{\pm\frac{1}{2}}$, *and test case 1 (uniform cube),* $m = 3$, $p = 3$ *(a) Multiquadric: The second column is the number of GMRES iterations, the third is the time it takes to achieve* $\epsilon \leqslant 10^{-3}$ *accuracy and the last is the total time to build the pre-conditioner, compute the HB basis and solve the iterations. (b) Inverse multiquadric: The column data is the same as for (a).*

model is now performed with different orders as shown in 4. The estimate and the Mean Square Error of the estimate on the unit cube $[0, 1]^3$ at $x_3 = 0.5$ is shown side by side for $m = 0$ (constant), $m = 1$ (linear) and $m = 2$ (quadratic) orders.

We can see in Figure 4(a) that with $m = 0$, the interpolated regression result deviates significantly from the underlying smooth signal $\|\vec{x}\|_1^3$. As we increase m, as shown in Figure 4(b)(c), we start to recover the smooth signal with $m = 2$, we have a recovered signal that is almost free of noise. The improvement as m increases is also evident from the rapid decrease of the expected point-wise mean squared error, as shown in the plots from the right column. All the BLUE estimate results were performed with the Kriging toolbox [33].

5. Conclusions. In this paper we construct a class of discrete HB that are adapted both to the RBF kernel function and the location of the interpolating nodes. The adapted basis has two main advantages: First the RBF problem is decoupled, thus solving the scale dependence between the polynomial and RBF interpolation. Second with a block SSOR scheme, or a simple diagonal matrix built from the multi-resolution matrix K_W, an effective pre-conditioner is built that reduces significantly the iteration count. Our result shows a promising approach for many RBF interpolation problems.

Further areas of interest as future work:

N	GMRES	GMRES residual	Total (s)
1000	48	1.48E-04	71
2000	95	6.71E-05	88
4000	134	3.16E-05	132
8000	150	2.71E-05	268
16000	274	1.85E-06	782

TABLE 4.6

Iteration and timing results for diagonal pre-conditioner, inverse multiquadric kernel $\phi(r) := 0.01(r^2 + 0.01^2)^{-\frac{1}{2}}$, *and Test Case 3 (bimodal Gaussian distributed centers),* $m = 3$, $p = 3$. *The second column is the number of GMRES iterations, the third is the GMRES residual needed to achieve* $\epsilon \leqslant 10^{-3}$ *accuracy, and the last is the total time to compute the HB basis, build the pre-conditioner and solve the iterations.*

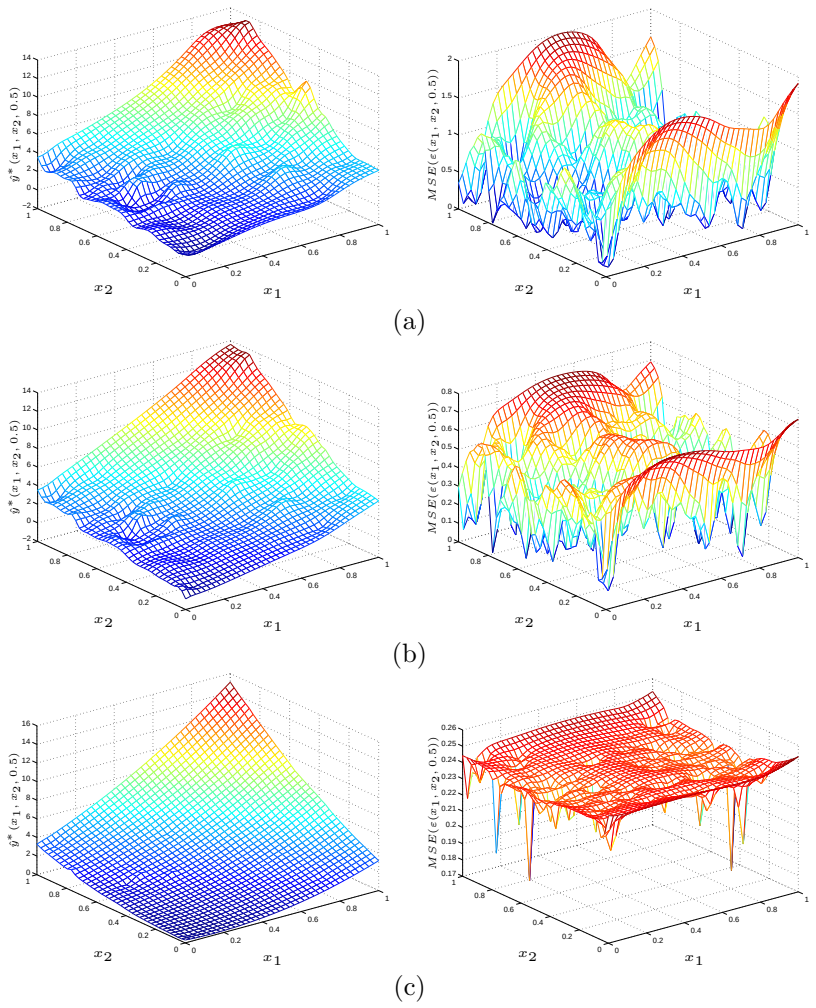


FIG. 4.2. *Interpolation results from the Kriging regression model with a correlation defined by the inverse multiquadric function. The figures in the left column show the interpolation result on a 40 by 40 mesh on the cross-section of a unit cube* $[0, 1]^3$ *at* $x_3 = 0.5$. *The figures in the right column show the point-wise mean squared error (MSE) on the sampling mesh. The results are computed for varied space* $\mathcal{P}^m(X)$ *of independent regression variables: (a)* $m = 0$ *(constant). (b)* $m = 1$ *(linear). (c)* $m = 2$ *(quadratic).*

- *Theoretical bounds of Convergence rates* The next step for the method that we have developed is to develop convergence rate estimates of the multi-resolution RBF matrix with the diagonal preconditioner.
- *Spatially varying anisotropic kernels.* An interesting observation is that the adapted discrete HB leads to a sparse multi-resolution RBF matrix representation for spatially varying kernels. This type of RBF interpolation has been gaining some interest lately due to the ability to better steer each local RBF function to increase accuracy. Due to the spatially varying kernel, we cannot use a fast summation method to optimally compute each matrix vector product. However, preliminary results show that we can sparsify the RBF matrix while retaining high accuracy of the solution. Full error bounds and numerical results will be described in a following paper that we are currently writing.
- *Extension to Integral Equations:* The approach described in this paper can also be applied to integral equations. The pre-conditioner would be built in the same fashion.

After we submitted this paper we became aware of the H-Matrix approach by Hackbusch [11] applied to stochastic capacitance extraction [61]. We have observed that the authors in this paper have built sparse $\mathcal{O}(N \log N)$ representations of the inverse operator for an electrostatic kernel. Although many (non-oscillatory) integral equation problems we are aware of have asymptotically decaying kernels, in principal this approach could be applied to RBF interpolation problems with increasing kernels that satisfy Assumption 1. Indeed, the approach for Gaussian Regression in [10] has many similarities with RBF interpolation.

There are many similarities between the H-Matrix and Hierarchical Basis approaches. The H-Matrix factorizes the operator matrix in a hierarchy of rectangular boxes and approximate each block with a low rank matrix. HB builds a multi-resolution basis with polynomial vanishing moments and sparsifies the matrix operator by applying a change of basis. Both approaches lead to sparse matrix representations and can be used for pre-conditioning [40, 41, 12].

One observation is that the H-matrix approach would still have the coupling between the polynomial and RBF components, which could still lead to a poorly conditioned system. One way around this problem is to combine the H-Matrix and HB approaches by decoupling the system with HBs and then applying the H-matrix to compute a sparse form of K_W^{-1}.

Acknowledgments. Thanks to Lexing Ying for providing a single processor version of the KIFMM3d code. We also appreciate the discussions, assistance and feedback from Raul Tempone, Robert Van De Gein, Vinay Siddavanahalli and the members of the Computational Visualization Center (Institute for Computational Engineering and Sciences) at the University of Texas at Austin. In addition, we appreciate the invaluable feedback from the reviewers of this paper.

REFERENCES

[1] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, *Wavelet-like bases for the fast solution of second-kind integral equations*, SIAM J. Sci. Comput., 14 (1993), pp. 159–184.

[2] B. K. Alpert, *A class of bases in l2 for the sparse representation of integral operators,*, SIAM J. Math. Anal., 24 (1993), pp. 246–262.

[3] K. Amaratunga and J. Castrillon-Candas, *Surface wavelets: a multiresolution signal processing tool for 3d computational modelling*, International Journal for Numerical Methods in Engineering, 52 (2001), pp. 239–271.

[4] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. http://www.mcs.anl.gov/petsc.

[5] R. Beatson, J. Cherrie, and D. Ragozin, *Fast evaluation of radial basis functions: Methods for four-dimensional polyharmonic splines*, SIAM J. Math. Analysis, 32 (2001), pp. 1272–1310.

[6] R. Beatson and L. Greengard, *A short course on fast multipole methods*, in Wavelets, Multilevel Methods, and Elliptic PDE's, M. Ainsworth, J. Levesly, W. Light, and M. Marietta, eds., Oxford Univ Press, 1997.

[7] R. K. Beatson, J. B. Cherrie, and C. T. Mouat., *Fast fitting of radial basis functions: Methods based on preconditioned gmres iteration*, Advances in Compuational Mathematics, 11 (1999), pp. 253–270.

[8] R K Beatson, W A Light, and Billings S, *Fast solution of the radial basis function interpolation equations: Domain decomposition methods*, SIAM J. Sci. Comput, 22 (2000), pp. 1717–1740.

[9] G. Beylkin, R. Coifman, and V. Rokhlin, *Fast wavelet transforms and numerical algorithms i*, Comm. Pure Appl. Math., 44 (1991), pp. 141–183.

[10] Steffen Börm and Jochen Garcke, *Approximating gaussian processes with h^2 matrices*, in ECML '07: Proceedings of the 18th European conference on Machine Learning, Berlin, Heidelberg, 2007, Springer-Verlag, pp. 42–53.

[11] S Borm, L. Grasedyck, and W Hackbusch, *Hierarchical matrices*, Lecture notes available at www.hmatrix.org/literature.html, (2003).

[12] Sabine Le Borne and Lars Grasedyck, *H-matrix preconditioners in convection-dominated problems*, SIAM Journal on Matrix Analysis and Applications, 27 (2006), pp. 1172–1183.

[13] J. W. Carnicer, W. Dahmen, and J. M. Pena, *Local decomposition of refinable spaces*, Appl. Comput. Harmon. Anal., 3 (1996), pp. 127–153.

[14] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and Evans T. R., *Reconstruction and representation of 3d objects with radial basis functions*, in SIGGRAPH 2001 proceedings, 2001, pp. 67–76.

[15] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell, *Smooth surface reconstruction from noisy range data*, in Proceedings of the 1st international conference on computer graphics and interactive techiques in Austalasia and South East Asia, 2003, pp. 119–126.

[16] G. Casciola, L.B. Lazzaro D., Montefusco, and S. Morigi, *Shape preserving surface reconstruction using locally anisotropic rbf interpolants*, Computers and Mathematics with Applications, 51 (2006), pp. 1185–1198.

[17] G. Casciola, L.B. Montefusco, and S. Morigi, *The regularizing properties of anisotropic radial basis functions*, Applied Mathematics and Computation, 190 (2007), pp. 1050–1062.

[18] J. Castrillon-Candas and K. Amaratunga, *Spatially adapted multiwavelets and sparse representation of integral operators on general geometries*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1530–1566.

[19] Julio E Castrillon-Candas and Jun Li, *Fast solver for radial basis function interpolation with anisotropic spattially varying kernels*, In preparation.

[20] J. Cherrie, R. Beatson, and G.N. Newsam, *Fast evaluation of radial basis functions: Methods for generalized multiquadrics in r^n*, SIAM J. Sci. Comput., 23 (2002), pp. 1549–1571.

[21] Yong Duan, *A note on the meshless method using radial basis functions*, Comput. Math. Appl., 55 (2008), pp. 66–75.

[22] Yong Duan and Yong-Ji Tan, *A meshless galerkin method for dirichlet problems using radial basis functions*, J. Comput. Appl. Math., 196 (2006), pp. 394–401.

[23] J. Duchon, *Splines minimizing rotation invariant semi-norms in sobolev spaces*, in Constructive Theory of Functions of Several Variables, Lecture Notes in Math., W. Schempp and K. Zeller, eds., Springer, Berlin, 1977, pp. 85–100.

[24] Richard. Franke, *Scattered data interpolation: Tests of some methods*, Mathematics of Computation, 38 (1982), pp. 181–201.

[25] Gene H. Golub and Charles F. Van Loan, *Matrix computations (3rd ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[26] L. Greenguard and V. Rokhlin, *A new version of the fast multipole method for the laplace equation in three dimensions*, Acta Numerica, 6 (1997), pp. 229–269.

[27] Nail A. Gumerov and Ramani Duraiswami, *Fast radial basis function interpolation via pre-*

conditioned krylov iteration, SIAM J. Sci. Comput., 29 (2007), pp. 1876–1899.

[28] Stefan Heedene, Kevin S. Amaratunga, and Julio E. Castrillon-Candas, *Generalized hierarchical bases: a wavelet-ritz-galerkin framework for lagrangian fem*, Engineering Computations, 22 (2005), pp. 15–37.

[29] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, JResNatBurStand, 49 (1952), pp. 409–436.

[30] Astrid Jourdan, *How to repair a second-order surface for computer experiments by kriging.*, Laboratoire de Mathématiques et de leurs Applications de Pau - LMA-PAU - CNRS : UMR5142 - Université de Pau et des Pays de l'Adour, (2007). 18 pages.

[31] L. Yu. Kolotilina and A. Yu. Yeremin, *Block ssor preconditionings for high order 3d fe systems*, BIT, 29 (1989), pp. 805–823.

[32] Soren N. Lophaven, Hans B. Nielsen, and Jacob Sondergaard, *Aspects of the matlab toolbox dace*, Tech. Report IMM-TR-2002-13, IMM, Informatics and Mathematical Moddeling. Technical University of Denmark, 2002.

[33] ———, *Dace: A matlab kriging toolbox*, Tech. Report IMM-TR-2002-12, IMM, Informatics and Mathematical Moddeling. Technical University of Denmark, August 2002.

[34] Jay D. Martin and Timothy W. Simpson, *A study on the use of kriging models to approximate deterministic computer models*, in Proceedings of DETC'04 ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 2004.

[35] ———, *Use of kriging models to approximate deterministic computer models*, AIAA Journal, 43 (2005), pp. 853–863.

[36] G. Matheron, *The intrinsic random functions and their applications*, Advances in Applied Probability, 5 (1973), pp. 439–468.

[37] C.A. Micchelli, *Interpolation of scattered data: Distance matrices and conditionally positive definite functions*, Constr. Approx., 2 (1986), pp. 11–22.

[38] Francis J. Narcowich and Joseph D. Ward, *Scattered-data interpolation on rn: Error estimates for radial basis and band-limited functions*, SIAM J. Math. Anal., 36 (2004), pp. 284–300.

[39] Hans Bruun Nielsen, *Surrogate models: Kriging, radial basis functions, etc.*, in Working Group on Matrix Computations and Statistics. Sixth workshop, Copenhagen - Denmark, April 1-3, 2005, ERCIM: European Research Consortium on Informatics and Mathematics, 2005.

[40] Suely Oliveira and Fang Yang, *H-matrix preconditioners for saddle-point systems from meshfree discretization*, Technical Report, University of Iowa.

[41] S. Oliveira and F. Yang, *An algebraic approach for h-matrix preconditioners*, Computing, 80 (2007), pp. 169–188.

[42] J.E. Pasciak, J.H. Bramble, and J. Xu, *Parallel multilevel preconditioners*, Math Comp., 55 (1990), pp. 1–22.

[43] T. von Petersdorff and C. Schwab, *Wavelet approximation for first kind integral equations on polygons*, Numer. Math., 74 (1996), pp. 479–516.

[44] ———, *Fully discrete multiscale galerkin bem*, in Multiscale Methods for PDEs, W. Dahmen, A Kurdila, and P. Oswald, eds., vol. 74, Academic Press,, San Diego, CA, 1997, pp. 287–346.

[45] Daniel Potts, Gabriele Steidl, and Arthur Nieslony, *Fast convolution with radial kernels at nonequispaced knots*, Numer. Math., 98 (2004), pp. 329–351.

[46] Yee P.V. and Haykin S., *Regularized Radial Basis Function Networks: Theory and Applications*, John Wiley, 2001.

[47] V J Romero, L P Swiler, and A A Giunta, *Application of finite-element, global polynomial, and kriging response surfaces in progressive lattice sampling designs*, in 8th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, 2000.

[48] Youcef Saad and Martin H Schultz, *Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[49] J Sacks, T.J. Welch, W.J. Mitchell, and H.P Wynn, *Design and analysis of computer experiments*, Statistical Science, 4 (1989), pp. 409–435.

[50] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Advances in Copmputational Mathematics, 3 (1995), pp. 251–264.

[51] ———, *Improved error bounds for scattered data interpolation by radial basis functions*, Mathematics of Computation, 68 (1999), pp. 201–216.

[52] P. Schroder and W. Sweldens, *Rendering Techniques: Spherical Wavelets: Texture Processing*, Springer Verlag, New York 1995, 95.

[53] Robin Sibson and G. Stone, *Computation of thin-plate splines*, SIAM J. Sci. Stat. Comput.,

12 (1991), pp. 1304–1313.

[54] Timothy W. Simpson, Timothy M. Mauery, John J. Korte, Multidisciplinary Optimization Branch, and Farrokh Mistree, *Comparison of response surface and kriging models for multidisciplinary design optimization*, in in AIAA paper 98-4758. 7 th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998, pp. 98–4755.

[55] Johannes Tausch and Jacob White, *Multiscale bases for the sparse representation of boundary integral operators on complex geometry*, SIAM J. Sci. Comput., 25 (2003), pp. 1610–1629.

[56] Z. Wu and R. Schaback, *Local error estimates for radial basis function interpolation of scattered data*, IMA Journal of Numerical Analysis, 13 (1993), pp. 13–27.

[57] Phannendra K. Yalavarthy, *A generalized least squares minimization method for near infrared diffuse optical tomography*, Ph.D Thesis Dartmouth College, 2007.

[58] P. K. Yalavarthy, B. W. Pogue, H. Dehghani, and K. D. Paulsen, *Weight-matrix structured regularization provides optimal generalized least-squares estimate in diffuse optical tomography*, Medical Physics, 34 (2007), pp. 2085–+.

[59] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole method in two and three dimensions*, Journal of Computational Physics, 196 (2004), pp. 591–626.

[60] Jun yong Noh, Douglas Fidaleo, and Ulrich Neumann, *Animated deformations with radial basis functions*, in VRST '00: Proceedings of the ACM symposium on Virtual reality software and technology, New York, NY, USA, 2000, ACM, pp. 166–174.

[61] Zhenhai Zhu and J. White, *Fastsies: a fast stochastic integral equation solver for modeling the rough surface effect*, Computer-Aided Design, International Conference on, 0 (2005), pp. 675–682.